UNIVERSITÉ DE MONTRÉAL

CENTRE DE RECHERCHE SUR LES TRANSPORTS

MÉMOIRE D'ÉTUDIANT

CASE POSTALE 6128, SUCCURSALE A, MONTRÉAL, QUÉBEC, H3C 3J7

CRT-605

FIABILITÉ ET ROUTAGE DANS UN RÉSEAU DE TÉLÉCOMMUNICATIONS

par

Brunilde Sansó

Thèse de doctorat, département de Mathématiques appliquées, École Polytechnique, dirigée par Monsieur François Soumis.

Université de Montréal
Centre de recherche sur les transports - Publication #605
Octobre 1988

Dépôt légal - 4e trimestre 1988

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE

Cette thèse intitulée:

FIABILITÉ ET ROUTAGE
DANS UN RÉSEAU DE TÉLÉCOMMUNICATIONS

présentée par: Brunilde Sansó
en vue de l'obtention du grade de: Philosophiae Doctor
à été dûment acceptée par le jury d'examen constitué de:

Monsieur Benjamin Smith, Ph.D., président
Monsieur Michel Gendreau, Ph.D.
Monsieur André Girard, Ph.D.
Monsieur François Soumis, Ph.D.

À Saúl et
David Eduardo.

" e intender non la puŏ
chi non la prova"

Dante.

RÉSUMÉ

Le problème traité dans cette thèse peut être divisé en deux parties. D'abord le choix d'une bonne mesure de fiabilité de réseau de télécommunications et, deuxièmement, l'étude de plusieurs modèles de routage d'appels.

Dans les travaux publiés en fiabilité de réseaux, les mesures de performance sont reliées à la notion de connexité du réseau. En télécommunications, à cause de la faible probabilité de panne des arcs et de la bonne connexité du réseau, la probabilité de déconnection est pratiquement nulle et ce type de mesures ne réflètent pas la vraie performance du réseau. De plus, la plupart des mesures de connexité nécessitent des algorithmes dont le temps d'exécution croit exponentiellement avec la taille du réseau, ce qui les rend difficiles à évaluer pour des réseaux de taille réelle.

Nous nous sommes donc attaqués d'abord au problème de la définition d'une mesure reflétant parfaitement la performance du réseau. La première contribution du travail est d'introduire une mesure de flot plutôt que de connexité. Le deuxième projet original à cette thèse est le fait que la méthodologie proposée pour le calcul de cette mesure relie les notions de fiabilité et de politique de routage, ce qui, à notre connaissance, est traité pour la première fois.

La mesure de fiabilité choisie est l'espérance du trafic acheminé dans le réseau. La méthodologie pour calculer cette mesure inclut l'énumération partielle des états les plus importants, l'évaluation du trafic sous une politique optimale de routage, lorsque il y a des pannes dans le réseau, et, finalement, des bornes supérieures et inférieures sur l'espérance du trafic.

Bien que le routage ne soit qu'un module à l'intérieur de notre méthodologie, nous avons développé plusieurs modèles de routage d'appels qui pourraient être utilisés même en dehors du contexte de fiabilité. Nous présentons des modèles qui obéissent à des politiques de routage centralisé ou décentralisé; nous prenons aussi en considération l'aspect stochastique de la demande.

Le modèle implanté reflète une politique centralisée de routage d'appels. Il s'agit d'un problème de multiflot où les contraintes de capacité des arcs ont été remplacées par l'ajout d'un terme d'espérance de pertes d'Erlang à la fonction objectif.

Nous mentionnons que la troisième contribution de cet étude est la démonstration de la convexité de la fonction espérance de pertes d'Erlang par rapport aux valeurs de demandes sur les arcs. La conjecture de la convexité de la fonction a été acceptée depuis longtemps en télécommunications, cependant, à notre connaissance, elle est prouvée ici pour la première fois.

Nous avons créé un ensemble de programmes pour évaluer la mesure de fiabilité choisie ainsi que d'autres paramètres du réseau. L'algorithme de base utilisé pour résoudre le problème de routage étant celui de Frank-Wolfe. Ce logiciel peut traiter en temps raisonnable des problèmes de taille réelle.

Même si l'objectif principal de ce travail est d'évaluer une mesure de fiabilité pour un réseau existant, nous esquissons dans la dernière partie comment utiliser la même méthodologie proposée pour les problèmes de synthèse de réseaux.

ABSTRACT

This thesis addresses two problems: firstly, the need to find a good reliability measure in a telecommunication network and, secondly, the study of several kinds of routing models.

Network reliability has been traditionally related to connectivity measures. Telecommunication networks are generally well connected and low in failures, therefore connectivity measures are usually very high even when the network is in a failure state. Moreover, most of the measures proposed up to now require algorithms that are solved in exponential time.

The first contribution of this work is related to the type of measure chosen to evaluate network performance. In fact, we have chosen the expected lost call traffic in a network under failure, a network flow measure rather than a connectivity one. The second contribution is the fact that for the first time, to our knowledge, a methodology has been developed to assess network performance under routing policies.

In order to calculate the expected lost call traffic, we propose to partially enumerate the most probable states of the system, to evaluate the traffic lost for every state considered and, finally, to use the Li and Silvester bounds to assess the measure of performance.

The outcome of the evaluation of the lost call traffic may differ according to the routing policy being used. We therefore propose several routing models to assess network performance. These models may be used even out of the reliability context. We have included centralized and decentralized policies in the proposed models.

The model actually implemented considers a decentralized routing policy and stochastic demands. The problem to be solved is a multicommodity flow one without capacity constraints. In fact, we have dropped these constraints since the objective function, the expected loss call traffic Erlang function, already penalizes arc overloads.

The third contribution of this thesis is the convexity proof of the expected lost call traffic Erlang formula relative to the offered traffic; which, to our knowledge, is shown here for the first time.

Finally, we point out that, even though the aim of this work was initially to assess the performance of an existing network, we have briefly stated how to use the same methodology for network design under reliability constraints.

REMERCIEMENTS

Les sentiments que j'ai éprouvés en écrivant cette partie de la thèse peuvent être résumés en deux mots: soulagement et gratitude. En effet, j'ai réalisé ce travail dans des circonstances tout à fait privilégiées; j'ai eu un directeur de recherche exceptionnel, j'ai été intégrée dans un environnement de recherche stimulant où il y a une excellente interaction entre chercheurs, personnel de soutien et ressources.

Je remercie donc, et avant tout, le premier responsable de cette situation, mon directeur, François Soumis. François, comme l'appelle son entourage, est tout ce qu'un étudiant gradué pourrait demander d'un directeur de thèse: très compétent et très disponible; il nous suit de très près, il nous pousse au bon moment, il donne le bon mot d'encouragement ou bien le terrible "c'est surprenant que" pour nous remettre dans la bonne voie. Il est, de plus, celui qui comprend les bonnes et les mauvaises petites situations de la vie. Merci, François.

J'aimerais remercier ceux qui, dans mon entourage, ont ressenti la même nécessité de démontrer quelque chose que l'on sait vrai; ceux qui ont joué la part de l'avocat du diable quand je faisais une conjecture, ceux qui ont critiqué et aimé mon travail.

Je pense avant tout à Michel Gendreau et à la chance que j'ai eue de profiter de son expérience. Je le remercie pour sa disponibilité quand tout allait bien et, aussi, quand tout allait mal.

La deuxième personne qui répond aux caractéristiques mentionnées est Pierre Trudeau. C'est lui qui a eu l'idée finale qui a abouti à la preuve de convexité qui avait consommé tant de notre temps. Merci, Pierre.

Je désire remercier les organismes qui ont contribués au financement de ma recherche et de mes études: B.N.R., CONICIT, le Ministère de l'Enseignement Supérieur et de la Science, les fonds F.C.A.R..

J'aimerais remercier tous ceux et celles qui, parmi les chercheurs, étudiants et personnel du Département de Mathématiques Appliquées, du C.R.T. et du GERAD, m'ont facilité la tâche de finir cette thèse.

Je mentionne, en particulier, le travail de Guy Brousseau, qui a tout fait pour mettre en place à temps le logiciel Graftek afin que je puisse réaliser mes graphiques. Les coups de main quotidiens de Pierre Lestage, qui entre autres, m'a introduit au monde merveilleux des C-shell. La disponibilité de Daniel Charbonneau pour toute question technique concernant le système. Le travail consciencieux et rapide de Lyne Bourbonnais et Sylvie Hétu au Centre de Documentation. L'extraordinaire efficacité du secrétariat du C.R.T.

L'algorithme de base de Frank-Wolfe a été programmé par Richard Thibault. J'aimerais le remercier pour l'avoir mis à ma disposition et pour ses suggestions et discussions lors des modifications nécessaires aux tests réalisés.

Lorsque l'on écrit une thèse dans une langue qui n'est pas la nôtre, on a toujours une difficulté et une insécurité de plus à surmonter. Le travail de lire et corriger est long et fastidieux. Je remercie ma bonne amie Lucie Bibeau pour avoir accepté de réaliser cette tâche dans des conditions loin d'être idéales et pour toujours avoir voulu me dépanner.

Je voudrais remercier André Langevin, mon compagnon de bureau, pour son encouragement et pour avoir partagé mes hauts et mes bas.

J'aimerais souligner le travail extraordinaire réalisé par Nicole Paradis. En effet, c'est grâce à sa persévérance que les symboles sont si beaux, les tableaux si concis et la présentation si soignée. C'est elle qui, avec patience et efficacité a pratiquement "édité" cette thèse. Merci, Nicole.

Finalement, je remercie les hommes de ma vie: David et Saúl. Le premier pour avoir accepté, avec philosophie, que sa maman soit un personnage drôle qui part tôt et arrive tard mais qu'on reçoit à bras ouvert avec le plus beau sourire du monde. Merci à Saúl pour son soutien de tous les jours, pour avoir joué le rôle de mère et de père lorsque mon travail le demandait, pour me mettre toujours en premier ... **Gracias.**

TABLE DES MATIÈRES

page

RÉSUMÉ vi

ABSTRACT ix

REMERCIEMENTS xi

TABLE DES MATIÈRES xiii

LISTE DES FIGURES xvi

INTRODUCTION 1

1. DÉFINITION DE FIABILITÉ 3
   - 1.1 Connexité 3
     - 1.1.1 Communication $s-t$ 3
     - 1.1.2 Communication entre tous les noeuds 4
     - 1.1.3 Espérances des mesures de connexité 5
   - 1.2 Introduction de la notion de flot 6
     - 1.2.1 Flot de s à t 6
     - 1.2.2 Flot de N à N 6
     - 1.2.3 Flot de s vers N 7
     - 1.2.4 Espérances des mesures de flot 8
   - 1.3 Autres types de mesures 8
     - 1.3.1 Nombre minimum d'arcs devant être enlevés des réseaux pour isoler un noeud du reste du graphe 8
     - 1.3.2 Nombre maximum de noeuds du système restant reliés 9
     - 1.3.3 Indice de fiabilité pondéré 9
     - 1.3.4 Espérances 10

2. LES MÉTHODES DE CALCUL 11
   - 2.1 Énumération complète 12
   - 2.2 Méthode des liens 14

2.3 Méthode des coupes 18
2.4 Algorithme de flot 21
2.5 Simulation 21
2.6 Énumération partielle 23
2.7 Méthodes de réduction, factorisation et modularisation 26
2.7.1 Réduction 26
2.7.2 Factorisation 28
2.7.3 Modularisation 29
2.8 Graphes particuliers 31

3. OBJECTIF ET MÉTHODOLOGIE 35
3.2.1 Énumération des m états les plus probables du système 36
3.2.2 Calcul de la demande non acheminée pour chacun des m états choisis du réseau 41
3.2.3 Calcul des bornes sur l'espérance du flot non acheminé dans le réseau 41

4. MODÈLES DE ROUTAGE 43
4.1 Nature du flot dans un réseau de télécommunications 45
4.2 Techniques de routage 46
4.3 Modèles de routage dans un réseau de télécommunications 48
4.3.1 Modèles d'optimisation centralisée 49
4.3.1.1 Modèles déterministes 49
4.3.1.2 Modèles stochastiques 56
4.3.2 Optimisation centralisée versus optimisation décentralisée 62
4.3.3 Modèles stochastiques décentralisés 66
4.3.3.1 Modèle hiérarchique (MS3) 68
4.3.3.2 Description de la fonction de débordement 71
4.3.3.3 Modèle de croissance proportionelle (MS4) 73
4.3.4 Équivalence entre les modèles proposés et des modèles d'optimisation globale 76
4.4 Algorithme de routage 80
4.4.1 Convergence de l'algorithme 81

5. RÉSULTATS 96
5.1 Présentation du réseau 96
5.2 Approximations de la fonction objectif 97
5.3 Résultats en routage 100
5.4 Résultats en fiabilité 112
5.4.1 Amélioration des bornes et conception de réseaux 121

CONCLUSION 126

BIBLIOGRAPHIE 129

LISTE DES FIGURES

		page
2.1	Réseau: exemple.	15
2.2 (a)	Remplacement parallèle.	27
2.2 (b)	Remplacement série.	27
2.3	Exemple de factorisation.	30
2.4	Exemples de graphes stochastiques.	32
3.1	Proportion de l'espace d'états couverts par les états de fonctionnement et de pannes simples.	39
3.2	Vue en détail de la figure 3.1.	40
4.1	Exemple qui montre le problème d'interdire certains flots si l'on utilise MD2.	55
4.2	Distribution de la probabilité du flot acheminé sur l'arc u.	57
4.3	Espérance du flot non acheminé sur l'arc u.	58
4.4	Réseau exemple.	66
4.5	Politique de routage du modèle MS3.	69
4.6	Répartition du flot sur les trois routes d'après le modèle MS3.	70
4.7	Politique du routage du modèle MS4.	74
4.8	Répartition du flot selon le modèle MS4.	75
5.1	Réseau "RED".	98
5.2	Qualité de l'approximation de $\mathcal{P}(X)$.	101
5.3	Organigramme des résultats pouvant être obtenus par les programmes de routage.	103
5.4	Temps d'exécution des programmes de routage vs proportion de la matrice de demande.	106
5.5	Temps d'exécution vs proportion de la matrice de demande pour un algorithme considérant un produit par noeud.	107
5.6	Moyenne du nombre d'arcs parcourus par unité de flot.	109
5.7	Moyenne du refus vs M.	111
5.8	Variance du % de refus vs M.	111
5.9	Exemple de la non-unicité d'une solution par produit.	113
5.10	Bornes du refus vs temps de défaillance.	115
5.11	Erreur de la borne supérieure du refus vs temps de défaillance.	118
5.12	Refus dû aux pannes vs temps de défaillances.	120
5.13	Comparaison des bornes du refus avec et sans réoptimisation.	124
5.14	Différences entre les bornes supérieures du refus avec et sans réoptimisation.	125

INTRODUCTION

Le problème de mesurer la fiabilité d'un réseau de télécommunications a été abordé de plusieurs façons. Dans la pluspart des travaux révisés, le réseau de télécommunications est représenté par un graphe probabiliste, c'est à dire un graphe dont les noeuds correspondent aux centres de commutation, et les arcs aux liaisons entre les centres de commutations. On associe une probabilité de panne à chaque arc. La majorité des auteurs évaluent la fiabilité comme une mesure de connexité du graphe. Cette approche, quoique pertinente pour un grand nombre de réseaux, n'est pas tout à fait appropriée pour les réseaux de télécommunications. En effet, étant donné que les composantes de ce type de réseaux ont généralement une faible probabilité de panne, la fiabilité, évaluée comme une mesure de connexité, ne montre pas nécessairement la véritable performance du réseau. De plus, les algorithmes existants pour calculer la plupart des mesures de connexité nécessitent des temps de calcul qui croissent exponentiellement avec la taille du réseau ou appartiennent au groupe de problèmes classifiés comme NP-difficiles.

Dans quelques travaux, on définit la fiabilité en termes de probabilité de pouvoir transmettre un certain flot. Cette approche, beaucoup plus adéquate que la précédente si l'on veut évaluer la performance du réseau, n'a pas cependant été exploitée de façon détaillée dans un contexte de fiabilité.

Le but principal de ce travail est donc de trouver une bonne mesure de fiabilité, c.-à-d. une mesure qui reflète la véritable performance du réseau et qui puisse être calculée en des temps raisonnables pour des réseaux de taille réelle. La mesure proposée ainsi que la méthodologie nécessaire à son calcul sont exposées au chapitre trois.

Les deux premiers chapitres sont consacrés à la révision des travaux publiés en fiabilité de réseaux. Le premier décrit les types de mesure de fiabilité utilisés par plusieurs auteurs alors que le deuxième traite des méthodes de calcul de ces mesures.

Lors de l'exposé de la méthodologie pour le calcul de la mesure de fiabilité proposée au troisième chapitre, nous montrerons comment notre objectif principal qui était d'évaluer la fiabilité du réseau nécessite la résolution de problèmes d'acheminement optimal d'appels. Nous avons donc consacré le quatrième chapitre aux modèles de routage . Nous y ferons, d'abord, un bref survol de la terminologie et des travaux pertinents et, par la suite, nous proposerons plusieurs modèles correspondant à plusieurs politiques de routage.

Il est à remarquer que les modèles de routage proposés sont intéressants en soi, c.-à-d. même en dehors du contexte de la fiabilié. En effet, les nouvelles technologies de commutation en télécommunications permettent plusieurs politiques d'acheminement d'appels. Il s'ensuit que les modèles que l'on proposera au chapitre quatre constituent un outil important pour évaluer la pertinence d'une politique en étude avant son implantation définitive dans le réseau.

Pour ce qui est de la fiabilité, nous étudierons le comportement du réseau pour une période de temps fixe, sans considérer les distributions des durées de vie des composantes ni les temps de réparation des pannes. Il est à noter cependant que bien que la mesure de fiabilité soit statique dans le temps, on introduira une notion de prise de décisions pour l'utilisation de modèles de routage optimal après pannes.

Dans le dernier chapitre, nous présenterons des résultats numériques relatifs à la fiabilité ainsi que ceux obtenus par l'implantation d'un des modèles de routage proposés au chapitre quatre.

1. DÉFINITION DE FIABILITÉ

Dans les études de systèmes physiques, les difficultés sont généralement reliées aux méthodes de mesure et non à la définition du paramètre à mesurer. Cependant, dans les réseaux de télécommunications, la fiabilité s'avère un paramètre difficile à définir et à mesurer, à cause des caractéristiques physiques de tels réseaux.

La probabilité de connexion entre deux noeuds est une mesure de fiabilité généralement acceptée pour un réseau. Dans un réseau de télécommunications cette mesure s'avère exceptionnellement élevée car les taux de pannes sont faibles et le réseau est assez connexe et de dimension importante. La fiabilité ainsi définie ne donne donc pas une idée de la vraie performance du réseau. Cette définition ne considère pas non plus le graphe comme un réseau avec flot; or chaque noeud a une certaine demande qui sera satisfaite ou pas selon la performance du système. Trois grands groupes de définition de connexité ont été traités dans les travaux antérieurs:

1.1 Connexité

1.1.1 Communication $s-t$

La plupart des travaux en fiabilité de réseaux utilisent $P_c[s, t]$ comme mesure à évaluer, $P_c[s, t]$ étant défini comme suit:

$$P_c[s, t] = \{ \text{ Probabilité de communication entre les noeuds } s \text{ et } t \}.$$

Le terme communication indique qu'il existe au moins un chemin entre s et t (voir par exemple Fratta et Montanari (1973); Hansler, Mc Auliffe et Wilkov (1974); Rai et Aggarwal (1978); Hagstron (1983); Provan et Ball (1984)). Il s'agit d'une bonne mesure de la fiabilité, mais elle présente certains désavantages. Les désavantages

physiques de cette mesure ont été énoncés dans l'introduction de cette section. A ces derniers il faut ajouter les difficultés de résolution d'un algorithme calculant $P_c[s, t]$. En effet, pour le réseau le plus simple qui soit, celui où seulement les arcs font défaillance avec une même probabilité p, l'expression à évaluer est

$$P_c[s, t] = \sum_{i=0}^{b} A_{s,t}(i)\,(1-p)^i\,p^{b-i} \tag{1.1}$$

où:

$A_{s,t}(i)$ = nombre de combinaisons de i arcs tels
que s'ils sont les seuls à fonctionner,
il y a un chemin entre s et t,

b = nombre d'arcs dans le réseau.

Le temps de calcul d'un algorithme évaluant $P_c[s, t]$ croît de façon exponentielle avec le nombre d'arcs dans le réseau.

1.1.2 Communication entre tous les noeuds

La deuxième mesure de connexité dans le réseau est $P[c]$, la probabilité de communication entre tous les noeuds (opérationnels) du réseau.

Cette mesure peut être évaluée en utilisant itérativement l'algorithme de calcul de $P_c[s, t]$ (Ball (1979)) ou en utilisant des méthodes de graphes. Dans le deuxième cas, il s'agit de trouver "l'arbre de recouvrement" du graphe (voir Frank, Khan et Kleinrock (1972); Aggarwal et Rai(1981); Hagstrom (1983)).

$P[c]$ est une mesure beaucoup plus complète que $P_c[s, t]$ du point de vue de la fiabilité d'un réseau de télécommunications. En effet, il est plus réaliste d'évaluer la bonne performance du réseau en fonction de la probabilité de connexion de tous les noeuds. De plus, il est possible de mesurer $P[c]$ dans un réseau

réduit, ce qui du point de vue pratique, est très utile pour une compagnie de télécommunications intéressée à connaître la probabilité que les noeuds "importants" du réseau soient connectés.

La fonction du réseau étant d'acheminer un flot d'appels, une bonne mesure de $P[c]$ ne garantit pas que la demande des noeuds connectés soit satisfaite.

1.1.3 Espérances des mesures de connexité

Des mesures moins utilisées sont traitées dans l'étude de Ball (1979). Il s'agit de l'espérance du nombre de paires de noeuds communiquants, de l'espérance du nombre de noeuds communiquant avec un certain noeud s, et de l'espérance du nombre de paires de noeuds pouvant communiquer avec s (voir aussi Hansler, Mc Auliffe et Wilkov (1974)).

$$E[ep] = E \text{ [nombre de paires de noeuds communiquant entre eux]},$$

$$E[en(s)] = E \text{ [nombre de noeuds communiquant avec } s\text{]},$$

$$E[ep(s)] = E \text{ [nombre de paires de noeuds communiquant avec } s\text{]}.$$

Ces mesures sont spécialement utiles dans des réseaux de communication militaires ou dans les réseaux d'ordinateur car la topologie et la nature des défaillances de ces réseaux font en sorte que les déconnections après pannes soient plus sévères que pour les réseaux de téléphone, par exemple.

Les temps de calcul des algorithmes pour évaluer $E[ep]$, $E[en(s)]$, $E[ep(s)]$ sont aussi exponentiels. La méthode de Ball donne des résultats intéressants pour des petits réseaux.

1.2 Introduction de la notion de flot

Le besoin de trouver d'autres mesures de fiabilité incluant les notions de demande, de capacité et de flot, nous force à redéfinir le réseau de télécommunications. Il ne s'agit plus d'un simple graphe probabiliste mais d'un réseau de transport stochastique. Nous avons alors un réseau avec flot dont les capacités des arcs sont connues et où la demande pour chaque paire de noeuds origine-destination est un processus de Poisson de paramètres variables dans le temps.

1.2.1 Flot de s à t

La définition de fiabilité la plus simple qui utilise la notion de flot est la probabilité de pouvoir transmettre un certain flot du noeud source s au noeud puits t. On retrouve cette définition dans les deux articles suivants: Lee (1980) et Aggarwal, Chopra et Bajwa (1982). L'algorithme de Lee utilise une notion d'ordre lexicographique combinée à l'algorithme de Ford et Fulkerson pour trouver le flot dans un graphe. Nous allons définir formellement la notion de lien à la Section 2.2. Il faut noter cependant qu'un lien entre deux noeuds correspond à un chemin qui relie ces deux noeuds. L'algorithme de Aggarwal utilise une matrice de liens avec un vecteur de capacité associée. L'idée de flot maximal est abordée de façon intuitive mais aucun algorithme de flot n'est utilisé.

Frank (1974b) utilise également l'algorithme de Ford et Fulkerson, mais l'originalité de son étude repose plus sur le type de réseau considéré que sur l'algorithme utilisé.

1.2.2 Flot de N à N

Une autre définition de fiabilité est la probabilité que pour toutes les paires de noeuds la demande soit acheminée. Cette définition est utilisée dans les articles de Doulliez et Jamouille (1972) et de A. Shogan (1982). Dans les deux cas, des méthodes de décomposition de réseaux sont proposées; cependant l'application de

telles méthodes ainsi que l'utilisation de la définition de fiabilité n'est faite que pour des réseaux abstraits (bien que Shogan fasse référence à des réseaux de transport d'énergie électrique).

L'application de la définition de fiabilité donnée ci-dessus aux réseaux des télécommunications n'est pas évidente. A la différence des réseaux électriques où l'on veut fournir à chaque noeud puits le même type d'énergie même si celle-ci provient de plusieurs sources, dans des réseaux de télécommunications il y a plusieurs types de services à fournir. Les appels provenant du noeud A devant être acheminés vers le noeud B constituent un produit, ceux du noeud B vers A sont un produit différent. Le nombre maximum de produits est ainsi le nombre d'ensembles ordonnés de paires de noeuds.

Trouver le flot acheminé sur le réseau est équivalent à résoudre un multi-flot compatible (à cause des capacités de chaque arc) (voir Gondran et Minoux). Hagstrom (1983) suggère l'application de son algorithme de décomposition (décrit plus en détail au chapitre deux) à des réseaux de transports en utilisant la définition de fiabilité donnée dans cette section. Son approche ne permet pas de traiter la double nature (source-puits) de chaque noeud, et à moins d'être utilisée dans des réseaux de transport à caractéristiques très spécifiques (autoroutes nord-sud par exemple), l'on pourrait penser que les flots s'annuleraient.

1.2.3 Flot de s vers N

Dans la section précédente, nous avons considéré le problème de transmission d'appels entre toutes les paires de noeuds du réseau. Il n'est pas inintéressant de traiter le problème comme un ensemble de problèmes de transmission de flot simple du noeud s à l'ensemble N. Il s'agit donc de considérer le produit "appels provenant du noeud s" et de trouver la probabilité de pouvoir transmettre à partir de s le flot désiré vers chacun des noeuds de N.

Ce type d'approche permettrait à la compagnie de télécommunications d'identifier les clients importants dans leur réseau et de baser sa mesure de performance sur le service offert à cet ensemble de clients. Cette mesure est évidemment

beaucoup moins coûteuse du point de vue du temps de calcul que la probabilité d'acheminer la demande de N vers N. En effet, dans ce dernier cas l'on traite $k \times k$ produits (k étant la cardinalité de N), tandis que le traitement "s vers N" requiert au plus k produits.

1.2.4 Espérances des mesures de flot

Des mesures de fiabilité plus raffinées n'ont pas encore été traitées dans la littérature, il s'agit des espérances de certains paramètres reliés au flot dans le graphe. Nous avons par exemple, l'espérance du flot maximal transmis de

- s à t,
- s à N,
- N à N.

Une mesure de même nature est l'espérance du flot non acheminé dans le graphe. Cette dernière sera considérée dans les chapitres 3 et 4.

1.3 Autres types de mesures

Nous regroupons dans cette section certaines mesures de type déterministe ou stochastique qui ne sont pas directement reliées à la connexité ni aux problèmes d'acheminement de flot. Ces mesures sont généralement une solution rapide à des problèmes pratiques qui peuvent se présenter dans différents types de réseau.

1.3.1 Nombre minimum d'arcs devant être enlevés des réseaux pour isoler un noeud du reste du graphe

Cette mesure traitée par Boesch et Thomas (1970) permet de trouver le nombre minimum d'arcs dont la défaillance isolerait un ensemble de m centres d'ordinateurs.

1.3.2 Nombre maximum de noeuds du système restant reliés

Cette mesure de fiabilité a été utilisée par Baran (1964) lors des études de résistance d'un réseau de télécommunications militaire après une attaque. Ce type d'approche serait aussi utilisable lorsque l'on est intéressé à une mesure de survivance après des événements catastrophiques. Une mesure encore plus spécifique a été utilisée (aussi dans des réseaux militaires) par Frank (1967). Il s'agit de l'espérance de la fraction de noeuds qui, après une attaque, peuvent communiquer avec un noeud choisi au hasard dans le réseau.

1.3.3 Indice de fiabilité pondéré

L'importance de la panne elle-même est un facteur à prendre en considération dans un réseau qui risque de tomber en panne. Dans un réseau de télécommunications, la perte d'une ligne à capacité presque infinie (fibre optique) est plus importante que celle d'une ligne à faible trafic. D'un autre côté, l'importance relative de l'arc en panne devrait aussi être considérée dans un indice de fiabilité: la perte d'une ligne reliant des services d'urgence (Pompiers, Police, Hôpital) est plus importante que la perte de ligne vers un centre essentiellement résidentiel.

Aggarwal (1985) utilise comme mesure de fiabilité la somme pondérée des probabilités de connexité de chaque lien entre s et t. La pondération est faite en divisant la capacité du lien par la capacité maximale entre s et t. L'article suggère aussi d'inclure la capacité requise entre s et t dans la pondération. Ainsi, si C_r est la capacité requise entre s et t, W_i et C_i le poids et la capacité du lien i respectivement, alors:

$$W_i = \begin{cases} C_i/C_r & \text{si} \quad C_i < C_r, \\ 1 & \text{sinon.} \end{cases}$$

Si la définition est très utile pour une compagnie de télécommunications, l'algorithme utilisé est cependant inefficace pour des grands réseaux. Nous en discuterons dans le chapitre 2 de ce travail. Bien que non mentionnée dans l'article,

nous suggérons l'utilisation de la même notion de pondération de fiabilité mais en prenant le poids comme une mesure d'importance de chaque arc. Cette mesure arbitraire pourrait évaluer, par exemple, l'importance d'un lien en fonction du type de client qui y est connecté aussi bien qu'en fonction de la capacité. On pourrait également tenir compte des revenus de chacune des lignes.

1.3.4 Espérances

Nous avons déjà mentionné plusieurs mesures utilisant des espérances au lieu des probabilités. Nous ajoutons cette section afin d'inclure les mesures d'espérances de n'importe quel paramètre énoncé par Li et Silvester (1984). Ils considèrent que si X est une mesure de performance du réseau dépendante des états de panne, son espérance $E[X]$ peut être approximée par une borne inférieure X_L et une borne supérieure X_M.

Des expressions généralisées sont données pour X_L et X_M et, ensuite, des exemples de mesures précises telles que la connexité $s-t$, le nombre de paires de noeuds communiquant entre eux et le délais dans le réseau sont évalués au moyen de formules.

La méthode de Li et Silvester est efficace non seulement du point de vue algorithmique mais aussi parce qu'elle laisse beaucoup de souplesse à l'utilisateur qui veut évaluer plusieurs mesures de fiabilité à la fois.

2. LES MÉTHODES DE CALCUL

Avant d'énumérer les méthodes de calcul existantes pour évaluer les définitions de fiabilité données dans le chapitre 1, nous allons expliquer les difficultés inhérentes à ce problème.

Premièrement, les méthodes de calcul de fiabilité de réseau nécessitent généralement des algorithmes dont le temps de calcul est exponentiel ou qui appartiennent au type de problèmes NP-difficiles. Ne voulant pas entrer dans les détails de la théorie de la complexité, nous nous limiterons à mentionner que lorsqu'un problème est défini comme NP-difficile, on ne peut pas assumer qu'il n'admette pas de solution en temps polynomial, on peut toutefois affirmer qu'il faudra au moins une découverte majeure pour le résoudre en un temps polynomial.

En deuxième lieu nous savons que la plupart des méthodes proposées sont limitées à des réseaux modélisés de façon simpliste ou "idéale" (l'on ne tient pas compte de toutes les caractéristiques physiques) et petits (exceptionellement certains articles traitent des réseaux de 20 noeuds).

Il faut aussi mentionner que beaucoup de recherches on été faites sur les réseaux d'ordinateurs plutôt que sur les réseaux de télécommunications ce qui rend plus tolérables les limitations expliquées ci-dessus.

Idéalement, la plupart des auteurs considèrent un réseau où les arcs tombent en panne de façon complètement indépendante. Cependant, il a été montré qu'une bonne partie des défaillances proviennent du partage d'équipement par certaines régions du réseau qui sont très souvent considérées disjointes. Dans des réseaux téléphoniques, la plupart des lignes partagent des câbles multipaires au moins entre chaque point terminal et le centre de commutation le plus proche. Ce qui fait que la plupart des arcs adjacents à un même noeud dans notre modèle devraient avoir des défaillances dépendantes.

Le fait de considérer les noeuds comme des éléments non défaillants est un autre point faible des modèles idéalisés. Cette hypothèse n'est pas réaliste, étant donné que les centres de commutation (les noeuds) sont dotés d'équipements électromécaniques et/ou électroniques sujets à des défaillances. La défaillance des

noeuds, considérée par plusieurs auteurs (voir Aggarwal, Gupta et Misra (1975); Ball (1979)) a été, elle aussi, modélisée de façon simplifiée: les noeuds, comme les arcs, ont des pannes de type "tout ou rien". En réalité, les noeuds peuvent fonctionner de façon détériorée, ils sont donc des éléments "multi-état" du point de vue de la fiabilité. De plus, les défaillances de certains circuits dans les noeuds entraînent des pannes pour les lignes qui y sont reliées. Il s'ensuit que les dépendances du réseau ne sont plus seulement entre arcs mais aussi du type noeud-arc.

Puisque plusieurs parmi les travaux antérieurs étudient des réseaux de communications militaires, il est pertinent de mentionner que le problème des dépendances est encore plus critique lorsque l'on mesure la fiabilité de ce type de réseau. En effet, ils utilisent un grand nombre de liaisons radio et ces liaisons peuvent être très affectées par les conditions météorologiques, l'interférence entre liaisons ou les effets de "chemins multiples": le signal à la réception est la somme de signaux déphasés ayant parcouru des chemins différents à cause du phénomène de réflexion. Nous savons donc que dans des réseaux de communications militaires, les dépendances existent non seulement entre liens adjacents, mais aussi entre ceux qui ne le sont pas.

Nous n'étudierons pas de façon séparée les travaux qui ont abordé les problèmes de dépendance; nous allons plutôt les incorporer dans les sections suivantes selon la méthode de calcul de fiabilité utilisée.

Nous pouvons indiquer que les travaux étudiés font toujours un compromis entre la complexité de la modélisation et la taille du réseau, car les algorithmes les plus efficaces ont été trouvés pour les réseaux définis comme "idéaux".

2.1 Énumération complète

Une façon simple d'évaluer une mesure de fiabilité d'un réseau est d'énumérer tous les états possibles du système et d'évaluer sa performance pour chaque état. Si la mesure de performance est une espérance d'un certain paramètre X et l'état du système est définit par l'état (panne ou fonctionnement) de ces composantes (arc ou noeuds), alors:

$$E[X] = \sum_{k=1}^{2^n} P(S_k)X(S_k) \qquad (2.1.1)$$

où

S_k = représentation du $k^{ième}$ état du système,
n = nombre d'éléments défaillants du système,
$P(S_k)$ = probabilité que le système soit dans l'état S_k.

Parfois la mesure de fiabilité est tout simplement la probabilité d'avoir le système en "bon fonctionnement". A ce moment là, si les états de "bon fonctionnement" sont connus et numérotés de 1 à m, alors

$$P[\text{ bon fonctionnement }] = \sum_{k=1}^{m} P(S_k).$$

Cette méthode permet d'évaluer la plupart des mesures de fiabilité d'un système, même si elles sont très complexes.

Les désavantages sont du point de vue de temps de calcul. Dès que le réseau atteint une certaine taille le nombre d'états (2^n) devient astronomique. Pour un réseau avec $n = 20$ composantes on a plus d'un million d'états, et un réseau de 100 composantes nécessite déjà plusieurs siècles de calcul.

Parmi les travaux étudiés, aucun article n'est basé sur l'énumération complète. Deux travaux, non inclus dans cette étude à cause des mesures dynamiques de fiabilité proposées, utilisent l'énumération complète. Il s'agit de Bonaventure et al. (1980) et de Meyer (1980). Dans les deux cas, le retard dans le système est calculé pour chaque état possible. Bonaventure et al. utilisent l'espérance du retard comme mesure de fiabilité tandis que Meyer calcule la distribution de probabilité du retard.

2.2 Méthode des liens

Nous appellerons lien entre s et t tout chemin qui relie le noeud s au noeud t. Supposons qu'un certain graphe possède m liens entre s et t et que $L_1, L_2, \ldots, L_m$ sont leurs ensembles associés, c.-à-d. l'ensemble des éléments qui constituent un chemin entre s et t.

Définisson S comme l'événement, "il existe au moins un chemin opérationel entre le noeud s et le noeud t". Alors, si E_i dénote l'événement, "tous les éléments du lien i fonctionnent", nous avons:

$$S = E_1 \cup E_2 \cup E_3 \cup \ldots \cup E_m. \tag{2.2.1}$$

Si la fiabilité F du système est définie comme étant la probabilité qu'il y ait une communication entre s et t, alors:

$$F = Pr\{S\} = Pr\{E_1 \cup E_2 \cup E_3 \cup \ldots \cup E_m\}. \tag{2.2.2}$$

L'équation précédente est relativement facile à évaluer si $E_1, E_2, \ldots, E_m$ sont des événements mutuellement exclusifs, en effet dans ce cas nous aurons:

$$F = Pr\{E_1\} + Pr\{E_2\} + \ldots Pr\{E_m\}. \tag{2.2.3}$$

L'évaluation de (2.2.2) se complique lorsque $E_1, E_2, \ldots, E_m$ ne sont pas des événements mutuellement exclusifs. Dans ce cas (2.2.2) serait évalué en utilisant le principe d'inclusion-exclusion (voir Feller(1968)).

$$F = S_1 - S_2 + S_3 - S_4 + \ldots (-1)^{m+1} S_m, \tag{2.2.4}$$

où

$$S_1 = \sum_{i=1}^{m} Pr\{E_i\}, \tag{2.2.5}$$

et, de façon généralisée

$$S_k = \sum_{1 \le i_1 < i_2 < \cdots < i_k \le m} Pr\{E_i \cap E_j \cap \ldots \cap E_k\}. \qquad (2.2.6)$$

Un lien minimum ou élémentaire est défini comme l'ensemble minimum d'éléments dont le fonctionnement assure la communication entre s et t. Sur le réseau de la figure 2.1, les e_i sont les éléments pouvant tomber en panne et $L_1 = \{e_1, e_4\}$, $L_2 = \{e_2, e_5\}$, $L_3 = \{e_1, e_3, e_5\}$ et $L_4 = \{e_2, e_3, e_4\}$ sont les liens minimaux entre s et t. Posons x_i = événement "l'élément e_i fonctionne", alors les événements associés au liens minimaux sont: $E_1 = \{x_1, x_4\}$, $E_2 = \{x_2, x_5\}$, $E_3 = \{x_1, x_3, x_5\}$ et $E_4 = \{x_2, x_3, x_4\}$.

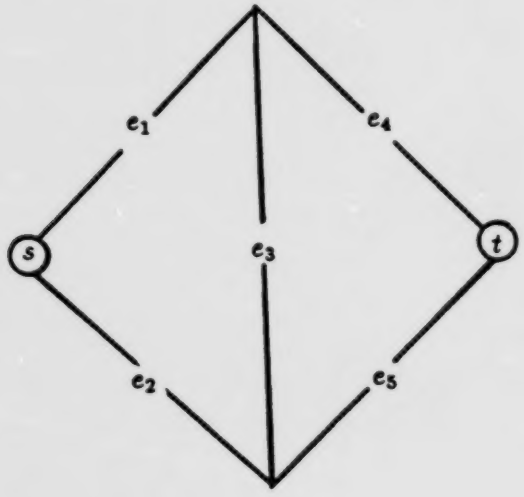


Figure 2.1 Réseau: exemple.

Supposons que chaque élément de la figure possède une probabilité de fonctionnement p_k et de panne $q_k = 1 - p_k$. L'application directe de la méthode de liens (formule (2.2.2)) pour calculer F n'est pas possible puisque les événements E_1, E_2, E_3 et E_4 ne sont pas mutuellement exclusifs. La formule (2.2.2) serait alors applicable uniquement aux liens modifiés, c.-à-d. aux liens représentants des événements mutuellement exclusifs.

Une façon de modifier les événements associés aux liens pour qu'ils représentent des événement mutuellement exclusifs consiste à agrandir l'ensemble et prendre en considération des éléments qui ne se trouvent pas dans le lien, l'on parle alors de lien modifié.

Ainsi, puisque x_k représente l'événement "l'élément k fonctionne", définissons $\overline{x}_k$ comme l'événement "l'élément k ne fonctionne pas." Il s'ensuit que S peut être exprimé par:

$$\begin{aligned} S = {} & \overline{x}_1 x_2 \overline{x}_3 \overline{x}_4 x_5 + \overline{x}_1 x_2 \overline{x}_3 x_4 x_5 \\ & + x_1 x_2 \overline{x}_3 \overline{x}_4 x_5 + x_1 x_2 \overline{x}_3 x_4 x_5 \\ & + x_1 x_2 \overline{x}_3 x_4 \overline{x}_5 + x_1 \overline{x}_2 \overline{x}_3 x_4 x_5 \\ & + x_1 \overline{x}_2 \overline{x}_3 x_4 \overline{x}_5 + \overline{x}_1 x_2 x_3 \overline{x}_4 x_5 \\ & + \overline{x}_1 x_2 x_3 x_4 \overline{x}_5 + \overline{x}_1 x_2 x_3 x_4 \overline{x}_5 \\ & + x_1 x_2 x_3 \overline{x}_4 x_5 + x_1 x_2 x_3 x_4 x_5 \\ & + x_1 x_2 x_3 x_4 \overline{x}_5 + x_1 \overline{x}_2 x_3 \overline{x}_4 x_5 \\ & + x_1 \overline{x}_2 x_3 x_4 x_5 + x_1 \overline{x}_2 x_3 x_4 \overline{x}_5. \end{aligned}$$

Par définition de F (eq. 2.2.2), l'expression de la fiabilité du système est identique à S après avoir substitué x_k par p_k et $\overline{x}_k$ par q_k.

Le calcul de F est aussi possible en prenant les liens élémentaires et en utilisant (2.2.4) à (2.2.6). Dans ce cas nous avons:

$$\begin{aligned} S_1 &= p_1 p_4 + p_2 p_5 + p_1 p_3 p_5 + p_2 p_3 p_4, \\ S_2 &= p_1 p_4 p_2 p_5 + p_1 p_4 p_3 p_5 \\ &\quad + p_1 p_2 p_3 p_4 + p_1 p_2 p_3 p_5 \\ &\quad + p_2 p_3 p_4 p_5 + p_1 p_2 p_3 p_4 p_5, \\ S_3 &= 4\, p_1 p_2 p_3 p_4 p_5, \\ S_4 &= p_1 p_2 p_3 p_4 p_5. \end{aligned}$$

Avec la première méthode nous avons 16 expressions à évaluer, dans le deuxième cas le nombre est de 15. Dans les deux cas le nombre croît exponentiellement avec les éléments du réseau.

Plusieurs auteurs ont essayé de développer des algorithmes simples travaillant avec des liens modifiés tout en simplifiant les expressions finales de fiabilité.

Fratta et Montanari (1973) proposent une réduction de la fonction de transfert $\emptyset(x)$ du graphe grâce à des manipulations booléennes. $\emptyset(x)$ est une fonction booléenne formée par une somme de produits; chaque produit correspond à un lien entre s et t. Les événements associés à chaque lien ne sont pas mutuellement exclusifs. Il s'agit donc de réduire $\emptyset(x)$ à chaque pas de l'algorithme, calculer la valeur arithmétique de fiabilité de façon itérative, sans passer par l'énumération de liens modifiés.

Le temps de calcul de l'algorithme est borné, pour le pire cas, par 3^n, cependant il est efficace pour des réseaux simples à faible connexité.

Une autre approche est donnée par Aggarwal, Gupta et Misra (1975), il s'agit de trouver des liens modifiés à partir des liens élémentaires. L'algorithme propose de construire séquentiellement les événements associés à L_k et qui sont mutuellement exclusifs de tous les événements énumérés depuis le début. Le temps de calcul de l'algorithme est toujours exponentiel. Cependant, la méthode proposée s'avère particulièrement efficace si le nombre de liens du graphe est assez élevé.

L'idée de transformer les événements associés aux liens élémentaires du graphe en événements mutuellement exclusifs est reprise par Rai et Aggarwal(1978). Ces auteurs proposent d'abord une procédure de recherche des liens élémentaires autre que l'inspection. La transformation des liens élémentaires de la fonction de transfert en liens modifiés est faite grâce à l'introduction de l'opérateur "exclusif". Bien que la méthode donne des temps de calcul exponentiel, le nombre de termes à manipuler est dans certains cas, inférieur de 40% par rapport aux algorithmes précédents.

Certains, parmi les articles révisés, utilisent la méthode des liens comme un simple outil, sans entrer dans les détails de perfectionnement algorithmique. Deux travaux traitent de l'évaluation d'une mesure de fiabilité dans un réseau de communications: Aggarwal, Chopra et Bajwa (1982) et Aggarwal (1985).

Le premier travail définit la fiabilité comme la probabilité de pouvoir transmettre un certain flot du noeud s au noeud t. Des liens valides sont ceux entre s et t dont l'ensemble des capacités est supérieure à la demande de flot requise. Un algorithme a été créé pour trouver les liens valides d'un réseau. Le travail est intéressant puisqu'il introduit la notion de flot dans la définition de fiabilité, mais l'algorithme nécessite de l'espace mémoire et des temps de calcul importants pour des réseaux de grande taille.

Le deuxième travail cité dans la section 1.3.3, introduit la notion de fiabilité pondérée par la capacité de chaque lien. Le désavantage de la méthode est la nécessité d'une énumération complète de tous les liens du réseau.

2.3 Méthode des coupes

L'ensemble d'éléments dont la défaillance déconnecte les noeuds s et t est appelé coupe entre s et t. Une coupe telle qu'il n'existe pas de sous-ensemble propre qui soit une coupe est une coupe minimale.

Reprenons l'exemple du pont montré dans la figure 2.1, alors si C_i dénote la $i^{ème}$ coupe minimale entre s et t nous avons:

$$C_1 = \{1, 2\}, \qquad C_2 = \{4, 5\}$$
$$C_3 = \{1, 3, 5\}, \quad C_4 = \{2, 3, 4\}.$$

Posons E_i = Événement "tous les éléments dans la coupe C_i sont en panne", alors l'événement de non communication entre s et t sera:

$$NC = E_1 \cup E_2 \cup E_3 \cup \ldots \cup E_n. \tag{2.3.1}$$

Posons P_f = Probabilité de non communication entre s et t. Alors F, la fiabilité du réseau sera:

$$F = 1 - P_f = 1 - Pr\{NC\}. \tag{2.3.2}$$

L'évaluation de $Pr\{NC\}$ n'est pas simple. En effet les E_i ne sont pas des événements mutuellement exclusifs d'où la nécessité d'utiliser le principe d'inclusion-exclusion (voir eq. 2.2.3 à 2.2.6) ou bien de transformer les coupes minimales de façon à ce que les E_i soient mutuellement exclusifs. Dans ce dernier cas nous aurions:

$$Pr\{NC\} = \sum_{i=1}^{n} P[E_i]. \tag{2.3.3}$$

Certains auteurs ont trouvé des méthodes pour modifier les coupes minimales de façon à ce que les événements associés soient mutuellement exclusifs. Le travail le plus cité est celui de Hansler, Mc Auliffe et Wilkov (1974) où ils proposent une procédure itérative pour calculer l'ensemble d'événements mutuellement exclusifs interprété comme coupe modifiée. Les besoins en espace-mémoire de l'algorithme croissent de façon polynomiale avec le nombre d'arcs du réseau; cependant le temps de calcul croît exponentiellement.

D'autres auteurs travaillent avec les coupes minimales et calculent des bornes sur la probabilité de la connexité entre s et t (voir Van Slyke et Frank (1972), et Cavers (1975)).

Pour des graphes dont le degré moyen est supérieur à quatre, le nombre de liens est beaucoup plus grand que celui de coupes. Dans ce cas les méthodes utilisant les coupes sont beaucoup plus rapides que celles basées sur les liens. C'est le cas des réseaux de télécommunications, d'où la popularité de la méthode des coupes dans la littérature étudiée.

Les travaux de Spragins, Assiri et Pan sont particulièrement originaux puisqu'ils modélisent le réseau de télécommunications dans sa forme la plus complète. Spragins (1977) donne une description des problèmes impliquant la dépendance des lignes et fait une analyse théorique de fiabilité dynamique. Spragins et

Assiri (1980) proposent un modèle appelé $q - \psi$ où les dépendances sont assez simplifiées: seules les probabilités de défaillance et les probabilités conditionnelles entre les arcs incidents à un même noeud sont considérées. La méthode des coupes modifiées citée au début de la section est utilisée pour le calcul de fiabilité. Pan et Spragins (1983) proposent un modèle plus détaillé. Il s'agit de considérer les dépendances entre les arcs, même s'ils sont non adjacents, d'utiliser la règle de la chaîne pour le calcul des probabilités de défaillance de chaque coupe tout en trouvant les coupes modifiées par la méthode déjà citée.

Les algorithmes de calcul de fiabilité qui considèrent les dépendances entre arcs sont intéressants du point de vue physique mais nécessitent des temps de calcul exorbitants. En effet, on a déjà vu que le temps de calcul des coupes modifiées d'un graphe est exponentiel; à ceci il faut ajouter la complexité de l'application de la règle d'enchaînement $\left(P(A \cap B) = P(A)\, P(B/A)\right)$ et le nombre d'événements dépendants à considérer. Prenons un cas où il y a m arcs auxquels l'arc l est correlé. Si chacun des m arcs peut être en fonctionnement ou en panne, il faut évaluer 2^m probabilités conditionnelles pour le seul arc l. Nous mentionnons, en relation aux problèmes de dépendance dans le réseau, qu'une nouvelle approche est donné par Lam et Li. Il s'agit du "modèle de réseau coloré". Tous les arcs ayant un même type de dépendance sont "colorés" avec la même couleur. Un arc peut avoir deux types de dépendance (un à chaque bout); de même, les arcs incidents à un même noeud peuvent être de couleurs différentes. Le modèle consiste alors à construire un sous-graphe pour chaque type de dépendance où tous les arcs d'un même sous-graphe tomberont en panne en même temps.

Provan et Ball (1984) ont développé un algorithme pour trouver la probabilité de connexion entre s et t. Dans leur algorithme l'énumération des coupes est toujours nécessaire, le temps de calcul croît de façon polynomiale avec le nombre de coupes $s - t$ du réseau. Ce résultat est important puisque le nombre de coupes croît de façon plus lente avec la taille du problème que d'autres types d'énumérations (tels que les liens).

2.4 Algorithme de flot

Tel que mentionné dans la section 1.2, certains auteurs ont utilisé la notion de flot pour définir la fiabilité du réseau (voir Doulliez et Jamoulle (1972); Lee (1980) et Shogan (1982)).

Tous les travaux cités utilisent l'algorithme de recherche de flot maximal de Ford et Fulkerson. Etant donné que l'algorithme est assez connu, nous ne mentionnerons que les références: Ford et Fulkerson (1974), et Gondran et Minoux (1979).

2.5 Simulation

Les méthodes de simulation étudiées ont été développées pour un graphe $G(N, A)$ où les noeuds sont généralement parfaitement fiables et où tout arc i fait défaut de façon aléatoire et indépendante avec une probabilité $1 - p_i$.

Supposons que l'on veuille connaître la probabilité de connexité entre les noeuds s et t du réseau. M_k représente le nombre de façons pour que s et t soient connectés lorsqu'il y a k arcs en opération et $n - k$ arcs en panne.

L'objectif des méthodes de simulation est d'estimer les valeurs de M_k au moyen d'échantillons.

Dans le cas où tous les arcs ont une même probabilité de fonctionnement:

$$p_i = p \quad \forall i \in A$$

la fonction de fiabilité $h(p)$ est définie par

$$h(p) = \sum_{k=1}^{n} M_k \, p^k \, (1-p)^{n-k} \tag{2.5.1}$$

où n est le nombre d'arcs du réseau.

Cependant, si les probabilités de fonctionnement varient pour chaque arc, il est nécessaire d'estimer des bornes sur la fonction de fiabilité; alors, pour $\underline{p} \leq p_i \leq \overline{p}$,

$$h(\underline{p}) \leq h(p) \leq h(\overline{p}).$$

Une méthode populaire de simulation est la stratification. Cette méthode est basée sur l'évaluation des termes de la fonction de défaillance plutôt que sur la fonction de fiabilité.

Posons:

q = probabilité de défaillance de chaque arc,

$p = 1 - q$ = probabilité de fonctionnement de chaque arc,

$d(p)$ = fonction de défaillance du réseau,

alors

$$d(p) = \sum_{k=1}^{n} C_k \, p^k \, q^{n-k}. \tag{2.5.2}$$

Si la fonction de défaillance est définie comme la probabilité de non connexité $s-t$ alors C_k représente le nombre de sous-réseaux contenant k arcs dans lequels s et t ne communiquent pas.

La stratification est basée sur le calcul des C_k. Les "strates" sont les réseaux sans défauts, avec 1 défaut, 2 défaut, et ainsi de suite. Chaque strate sera échantillonnée un nombre de fois proportionnel à sa probabilité d'avoir lieu. Certaines strates n'ont pas besoin d'être échantillonnées puisque l'on connait déjà leur valeur. Nous savons par exemple que

$$C_k = 0 \quad \text{si } k = n - \tau_{s-t} + 1, \ldots n$$

où τ_{s-t} est la grandeur (nombre d'arcs) de la coupe minimale séparant s et t.

Plusieurs travaux utilisent la simulation de Monte Carlo pour évaluer $h(p)$ ou $d(p)$. Le principe est toujours celui décrit précédemment, les différences consistent dans des raffinements du processus d'échantillonnage qui permettent une simulation plus rapide. (Voir Van Slyke et Frank (1972); Frank (1974b); Fishman (1985)).

2.6 Énumération partielle

Dans la section 2.1 nous avons vu la base des méthodes d'énumération. L'énumération complète, expliquée dans la section 2.1, présentait le désavantage du nombre d'états impliqués (donc du temps de calcul requis). Un deuxième désavantage nous vient à l'esprit: l'énumération complète prévoit le calcul de probabilité de tous les états, même de ceux qui sont très improbables. L'introduction de ces états dans l'étude du système non seulement n'est pas significative pour les mesures de fiabilité (gaspillage de ressources) mais, dans certains cas, nécessite des calculs en double précision qui ralentiront l'algorithme et créeront des problèmes de représentation de nombres trop petits dans l'ordinateur.

La méthode d'énumération partielle est basée sur l'énumération des états les plus probables. La valeur de performance à évaluer sera alors restreinte par des bornes supérieure et inférieure.

En reprenant la nomenclature de la section 2.1:

$S_k = k^{ième}$ l'état du système,

$P(S_k) =$ probabilité que le système soit dans l'état S_k,

$n =$ nombre d'éléments défaillants du système,

p_i = probabilité de bon fonctionnement de la $i^{ème}$ composante,

$q_i = 1 - p_i$ = probabilité de défaillance de la $i^{ème}$ composante.

La probabilité que le système soit dans l'état k peut s'exprimer par:

$$P(S_k) = \prod_{i=1}^{n} p_i \left(\frac{q_i}{p_i}\right)^{T_i(S_k)} \tag{2.6.1}$$

où

$$T_i(S_k) = \begin{cases} 0 & \text{si la composante } i \text{ fonctionne} \\ & \text{dans l'état } S_k, \\ 1 & \text{sinon.} \end{cases}$$

Un algorithme d'ordonnancement des états S_k par ordre de probabilité décroissante a été proposé par Li et Silvester en prenant la restriction $\frac{1}{2} \leq p_i \leq 1$.

Posons $R_i = q_i/p_i$ alors les composantes sont renumérotées par ordre croissant de R_i $(R_1 \geq R_2 \geq R_3 \cdots \geq R_n)$ et 2.6.1 peut être réécrite comme

$$P(S_k) = \prod_{i=1}^{n} p_i \, (R_i)^{T_i(S_k)} . \tag{2.6.2}$$

L'algorithme décrit utilise la notion suivante:

si $\left(P(S_a) \geq P(S_b) \geq \cdots \geq P(S_c)\right)$ et que $A = \{S_a, S_b, \ldots, Sc\}$ est un ensemble ordonné, alors $B = \left\{S_a \cup \{i\}, S_b \cup \{i\}, \ldots, S_c \cup \{i\}\right\}$ sera aussi un ensemble ordonné.

Note: $S_k \cup \{i\}$ correspond à un nouvel état dont la seule différence avec S_k est le fait que la composante i est aussi en panne.

La mesure de fiabilité traitée par cette méthode serait l'espérance $E[X]$ d'un certain paramètre X. Li et Silvester proposent des bornes supérieure X_M et inférieure X_L. Ces bornes sont données par (2.6.3) lorsque le paramètre X est croissant avec les pannes.

$$X_M = E[X]_{sup} = \sum_{k=1}^{m} P(S_k)\,X(S_k) + \left(1 - \sum_{k=1}^{m} P(S_k)\right) X(S_{2^n})$$

$$X_L = E[X]_{inf} = \sum_{k=1}^{m} P(S_k)\,X(S_k) + \left(1 - \sum_{k=1}^{m} P(S_k)\right) X(S_1) \qquad (2.6.3)$$

où :

m = nombre d'états considérés,

S_1 = état de bon fonctionnement de tous les éléments du système,

S_{2^n} = état de non-fonctionnement de tous les éléments du système.

L'algorithme de Li et de Silvester présente certains avantages de l'énumération complète (connaissance de ce qui arrive dans les états les plus probables du système) sans toutefois nécessiter un temps trop grand de calcul. En effet, la plus grande partie de l'espace d'état du système peut être couverte en temps polynomial par énumération partielle, comparé à temps exponentiel par énumération complète.

Il est intéressant de remarquer que si la probabilité de défaillance des arcs est petite, comme dans le cas des réseaux de télécommunications, les états de défaillance simple seront beaucoup plus probables que ceux de défaillances multiples. Il s'ensuit qu'il suffirait d'énumérer les états d'après l'ordre des probabilités de panne sur les arcs. Par contre, lorsque les probabilités de défaillance sur

les arcs sont très différentes, des états de défaillances multiples pourraient être plus probables que ceux de défaillances simples, d'où la nécessité d'un algorithme d'énumération des états.

2.7 Méthodes de réduction, factorisation et modularisation

2.7.1 Réduction

L'idée de la réduction est celle de simplifier un graphe au maximum avant l'application d'un algorithme de fiabilité quelconque. La réduction et le remplacement sont fortement liés, nous allons par la suite définir les deux concepts.

Un remplacement est une transformation topologique du graphe $G(V, E)$. Une réduction est un concept probabiliste puisqu'il regroupe le remplacement topologique ainsi que le calcul de mesures équivalentes de fiabilité dans un graphe stochastique, c.-à-d. un graphe dont les arcs ont une probabilité de panne.

Définissons la fiabilité $R(G_k)$ comme la probabilité de connexité des noeuds dans un sous-ensemble K de V, alors des remplacements séries et parallèles notés par le symbole $\sim$ et $\|$ respectivement, sont montrés dans les figures 2.2 (a) et 2.2 (b).

Afin de diminuer la taille de G_k et de ainsi réduire la complexité de calcul de $R(G_k)$, l'on applique des réductions qui préservent la fiabilité du graphe. Ces réductions altèrent le sous-graphe G_k du point de vue topologique et probabiliste pour obtenir G'_k de façon à ce que $R(G_k) = \Omega R(G'_k)$, Ω étant une constante qui découle du type de réduction introduite. À titre illustratif, définissons les trois réductions les plus courantes.

<u>Réduction parallèle</u>
Soit $e_a = (u, v)$ et $e_b = (u, v)$ deux arcs parallèles de G, alors si $p_a = 1 - q_a$ et $p_b = 1 - q_b$ sont les fiabilités respectives de e_a et e_b et que $e_c = e_a \| e_b$ on a que la fiabilité de e_c est donnée par $p_c = 1 - q_a q_b$, le facteur $\Omega = 1$ et $K' = K$.

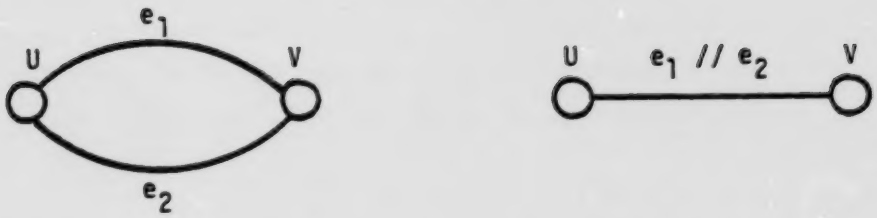


Figure 2.2 (a) Remplacement parallèle.

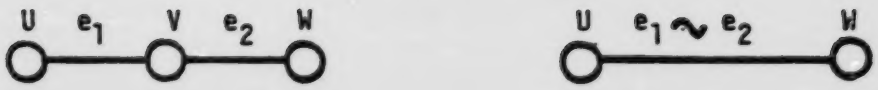


Figure 2.2 (b) Remplacement série.

Réduction série

Soit $e_a = (u, v)$ et $e_b = (v, w)$ deux arcs en série dans G tels que $\deg(v) = 2$ et $v \notin K$, alors $e_c = e_a \sim e_b$ et $p_c = p_a p_b$, $\Omega = 1$ et $K' = K$.

Réduction de degré 2

Soit $e_a = (u, v)$ et $e_b = (v, w)$ deux arcs en série dans G avec $\deg(v) = 2$ et $u, v, w \in K$. La réduction de degré 2 des arcs e_a et e_b produit un remplacement en série des arcs e_a et e_b par l'arc e_c tel que la fiabilité p_c de l'arc est donnée par $p_c = p_a p_b/(1-q_a q_b)$, ce qui implique que $\Omega = (1-q_a q_b)$ et $K' = K-v$.

2.7.2 Factorisation

La factorisation est une technique de décomposition d'un graphe en sous-graphes moins compliqués. La technique nécessite des réductions telles que celles montrées en 2.7.1, avant et pendant l'application des algorithmes de résolution.

Les algorithmes de factorisation existants ont été révisés par Wood (1986). Wood a basé son étude sur le calcul de fiabilité définie comme la probabilité de connexion d'un sous-ensemble K de noeuds. Les algorithmes révisés appliquent itérativement la formule suivante:

$$R(G_k) = p_i R(G_k * e_i) + q_i R(G_k - e_i) \qquad (2.7.1)$$

où :

$R(G_k)$ = fiabilité telle que définie dans la section antérieure,

$G_k * e_i = G_k$ avec e_i contracté,

$G_k - e_i = G_k$ avec e_i effacé,

$p_i = 1 - q_i$ = fiabilité de e_i.

Pour bien saisir l'idée de la factorisation, nous allons montrer un exemple dans la figure 2.3. Nous sommes intéressés à trouver la probabilité de connexion des deux noeuds en noir.

La fiabilité du graphe G_k montré dans la figure 2.3 est alors calculée par la formule 2.7.1 et donne:

$$\begin{aligned} R(G_k) = p_4\{&(1-q_3\,q_5)\,(1-q_1\,q_2)\,(1-q_6\,q_7) \\ &+ q_3\,q_5(p_1\,p_7 + p_2\,p_6 - p_1\,p_2\,p_6\,p_7)\}+ \\ &+ q_4\{(p_1\,p_3 + p_2 - p_1\,p_2\,p_3)\,(p_5\,p_7 + p_6 - p_5\,p_6\,p_7)\}. \end{aligned}$$

Il est assez évident que les algorithmes de factorisation donnent des temps de calcul exponentiel dans le pire des cas. Par rapport aux autres méthodes déjà révisées (énumération complète, approches booléennes de calcul de liens et de coupes), la factorisation requiert au plus, du stockage quadratique par rapport à la dimension du réseau (si l'on demande un résultat numérique seulement). Les autres méthodes requièrent généralement du stockage qui croît exponentiellement avec la taille du réseau.

2.7.3 Modularisation

La modularisation est une méthode de décomposition d'un réseau en un ensemble de sous-réseaux. Si la solution du problème de fiabilité d'un réseau de n composantes est donnée en temps $f(n)$, f est au moins exponentiel en fonction de n. Cependant la modularisation réduit le temps de calcul $f(n)$ en une somme de fonctions $f(n_i)$ où $n_i < n$. En d'autres mots, cette méthode profite de la structure particulière du réseau pour réduire le temps de calcul.

Hagstron (1983) propose de décomposer le graphe en sous-graphes rattachés sous forme d'arbre, appelé "arbre de décomposition". L'arbre de décomposition est trouvé en cherchant les composantes triconnexes du graphe. Hopcroft et Tarjan (1973) ont présenté un algorithme pour trouver les composantes triconnexes d'un graphe dont le temps de calcul croît linéairement avec la taille du réseau.

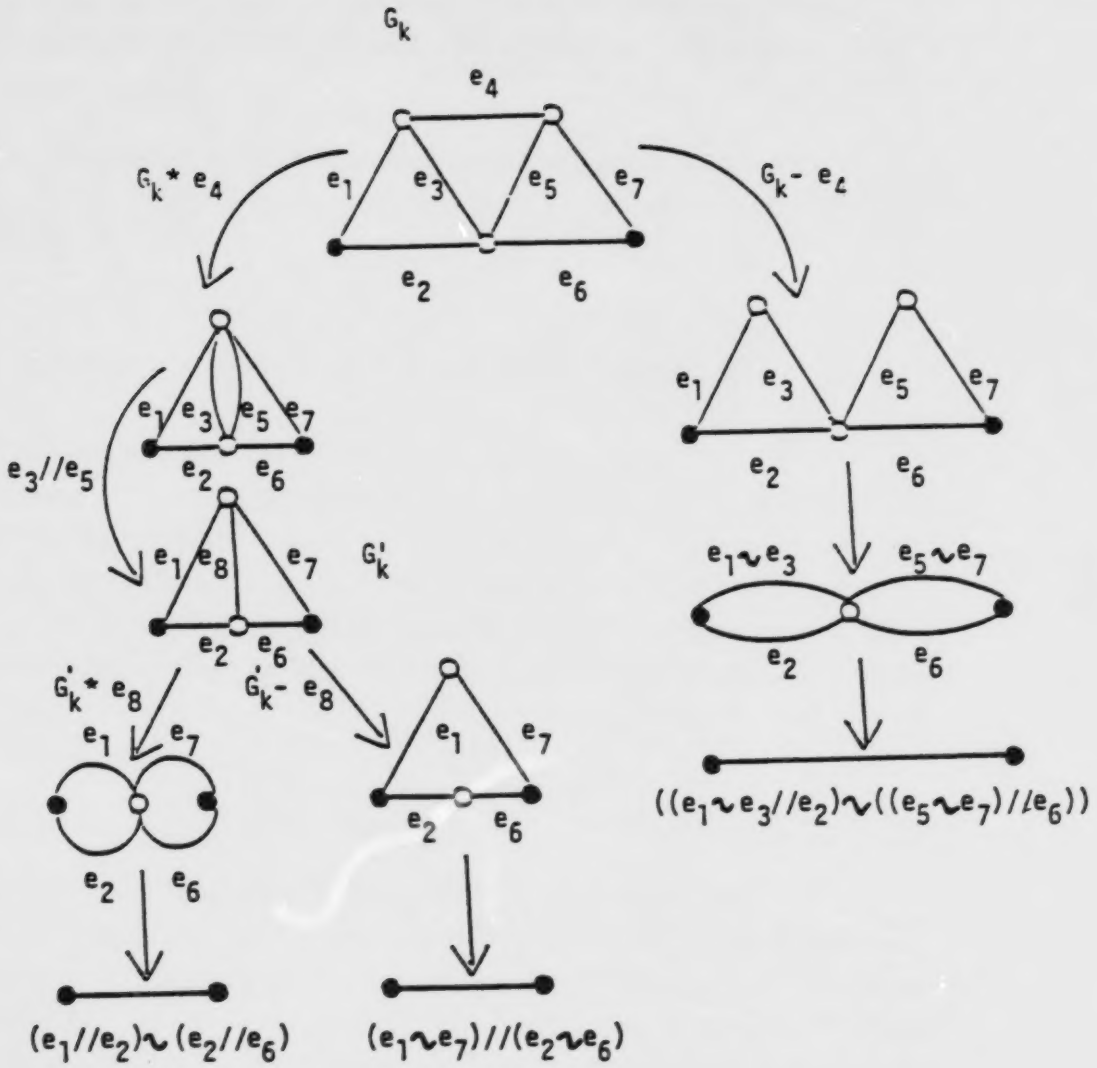


Figure 2.3 Exemple de factorisation.

Une fois créé l'arbre de décomposition, le calcul de la fiabilité se fait en étudiant la fiabilité de chacun des modules qui composent l'arbre à l'aide d'une méthode de factorisation. Il faut, évidemment, commencer par une feuille de l'arbre et calculer, à chaque fois, la probabilité de survie de l'arc en commun de cette feuille avec son "module parent". Une fois que ce paramètre est calculé le "module parent" sera la nouvelle feuille de l'arbre et le processus de calcul doit recommencer. Le calcul prend fin lorsque l'on calcule la fiabilité du "module racine" de l'arbre qui contient les noeuds dont on veut connaître la probabilité de connexité. Pour ce qui est de la complexité du calcul, vu que la méthode de factorisation à l'intérieur de chaque module est toujours utilisée, les temps de calcul croissent de façon exponentielle avec la taille du sous-réseau à l'intérieur du module. Cependant, comme on l'a déjà mentionné, l'efficacité de la méthode réside dans le fait d'avoir réduit un calcul exponentiel sur la totalité du réseau à une somme de calculs exponentiels dans des petits réseaux où le nombre de sous-réseaux croit linéairement avec la taille du graphe. La méthode est donc plus pertinente si le nombre de sous-réseaux est élevé car cela implique que le calcul de fiabilité se fera à l'intérieur de très petits réseaux.

Bien que l'on vienne de décrire l'algorithme qui trouve la probabilité de connexion $s-t$, Hagstron a montré qu'il est possible d'utiliser la même procédure pour des mesures de fiabilité différentes: communication entre tous les noeuds, probabilité de transport de flot. Le calcul de cette dernière mesure est cependant limité aux réseaux orientés.

2.8 Graphes particuliers

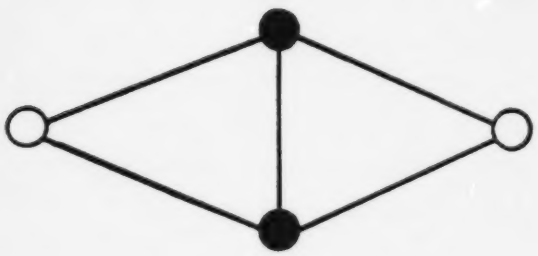
Bien que le calcul exact des mesures de fiabilité de réseaux nécessite, presque toujours, un temps de calcul exponentiel, il existe une certaine famille de graphes pour lesquels des algorithmes polynomiaux ont été développés. Il s'agit de certains graphes planaires, en particulier les graphes série-parallèle. Politof et Satyanarayama (1986) ont fait une synthèse des algorithmes polynomiaux existants dans ce domaine.

Un graphe non-orienté est série-parallèle s'il peut être réduit à un seul arc suite à des remplacements récursifs séries et parallèles. Un graphe stochastique est série-parallèle si le graphe correspondant est série-parallèle.

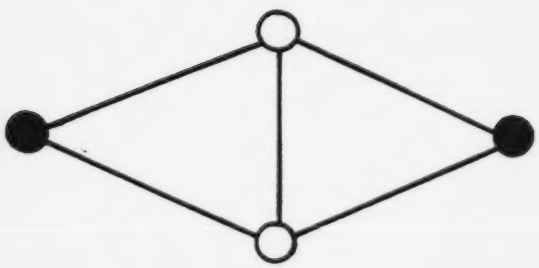
Parmi les graphes stochastiques série-parallèle l'on peut distinguer deux types:

sp-réductibles ("*sp*-reducibles")
sp-complexes ("*sp*-complex").

Un graphe stochastique est *sp*-réductible s'il peut être simplifié jusqu'à obtenir un arc par réductions séries, parallèles et de degré 2. Si ceci n'est pas possible, le graphe est *sp*-complexe. Dans la figure 2.4 (a) et (b) nous montrons un example de graphe *sp*-réductible et *sp*-complexe respectivement:



(a) graphe *sp*-réductible



(b) graphe *sp*-complexe

Figure 2.4 Exemples de graphes stochastiques.

En fonction des définitions montrées ci-dessus, il est possible de dire que la fiabilité (probabilité de connexité entre les noeuds du sous-ensemble K) est calculable en temps polynomial par des réductions série-parallèle si le graphe est *sp*-réductible (voir les travaux cités par Politoff et Satynarayana).

Pour ce qui est des graphes *sp*-complexes, de nouvelles propriétés ont été récemment découvertes. La propriété la plus marquante est la réduction d'un polygone à une chaîne. Une chaîne est une succession de noeuds qui commence et finit avec des noeuds et dont les noeuds intermédiaires sont de degré 2. Un polygone est l'union de deux chaînes qui ont le même noeud initial et le même noeud final.

Satyanarayana et Wood (1985) ont démontré que si un graphe est *sp*-complexe, il admet au moins un polygone. Ce dernier doit faire partie d'un des sept types décrits et il y aura une réduction polygone-chaîne possible. Ces réductions sont effectuées en temps constant. Deuxièmement l'article montre que pour trouver un polygone dans le graphe il est nécessaire de chercher trois chaînes à partir de chaque noeud, la recherche prend alors au plus $\mathcal{O}(V)$. Finalement la réduction chaîne-polygone permet la transformation du graphe *sp*-complexe en *sp*-réductible pour lequel il y a des algorithmes polynomiaux de résolution.

Une approche différente pour trouver des algorithmes polynomiaux pour des graphes *sp*-complexes est donnée par Wald et Coulbourn (1983). Ils transforment le graphe en une généralisation des graphes série-parallèle appelé 2-arbre. Un 2-arbre est défini de façon récursive comme suit:

1) un arc simple est un 2-arbre.

2) étant donné un 2-arbre G et $\{u, v\}$ un arc de G, si l'on ajoute un nouveau noeud w et des arcs $\{u, w\}$ et $\{v, w\}$, le graphe résultant est un 2-arbre.

Lorsque cela est possible, la transformation du graphe original (*sp*-complexe) en un 2-arbre est réalisée par l'ajout des arcs avec fiabilité nulle; ceci est accompli en temps linéaire. La deuxième phase de l'algorithme consiste à calculer six mesures de fiabilité pour chaque arc et subséquemment réduire le graphe

agrandi par des opérations série-parallèle et de degré-2. Cette tâche est aussi accomplie en temps polynomial. En conclusion, des algorithmes polynomiaux ont été trouvés pour tous les graphes série-parallèle et pour une généralisation des série-parallèles appellées 2-arbres.

Les graphes série-parallèle appartiennent à une famille de graphes planaires appelées "Inner Cycle Free" ou ICF. Nous n'allons pas entrer dans les définitons des autres groupes de graphes ICF puisque ceci dépasserait le but de ce travail. Nous nous limitons à noter qu'il existe des algorithmes polynomiaux pour résoudre des graphes ICF et qu'ils ont été décrits par Politof et Satyanarayana dans l'article cité au début de la section.

3. OBJECTIF ET MÉTHODOLOGIE

Le but principal de notre travail est de trouver une mesure de fiabilité (ou de non fiabilité) qui reflète la réalité et qui puisse être calculée en un temps raisonnable.

On aurait pu envisager la probabilité de connexion entre deux ou plusieurs noeuds du réseau comme mesure de fiabilité. Toutefois elle n'est pas considérée comme une bonne mesure pour les raisons suivantes:

- Toutes les mesures de connexité sont très coûteuses à trouver pour un réseau de taille importante;

- La connexité d'un réseau de télécommunications est toujours assez élevée même après plusieurs pannes. Il s'ensuit que la probabilité de connexion entre noeuds est toujours proche de 1 et qu'elle n'est donc pas un bon indice de la performance du réseau;

- L'importance des clients déconnectés à cause d'une panne ainsi que la nature (stochastique ou déterministe) de la demande ne sont pas considérés.

Nous avons donc envisagé une mesure de fiabilité pouvant tenir compte des problèmes décrits ci-dessus. Il s'agit d'une valeur pondérée de l'espérance du flot non acheminé dans le réseau dû uniquement aux pannes. Afin de trouver telle mesure on doit suivre les étapes suivantes:

3.1) Trouver la valeur du flot pondéré non acheminé dans le réseau lorsqu'il n'y a pas de défaillance.

3.2) Trouver l'espérance du flot non acheminé dans le réseau lorsqu'il y a des pannes. Ceci sera l'objet d'une discussion supplémentaire et constitue le noyau de ce travail.

3.3) Si l'on soustrait la valeur trouvée en 3.2) de celle obtenue en 3.1), on trouvera notre mesure de fiabilité: l'espérance du flot non acheminé dû uniquement aux pannes pouvant survenir dans le réseau.

La formulation pour trouver le routage optimal, lorsqu'il n'y a pas de défaillance dans le réseau, est similaire aux modèles qui seront expliqués au chapitre 4. Pour ce qui est de trouver l'espérance du flot non acheminé en présence de pannes (point 3.2) nous avons envisagé la procédure suivante:

3.2.1 Etant donné un réseau dont seuls les arcs peuvent faire défaillance, utiliser un algorithme d'énumération partielle d'états (celui de Li et Silvester, par exemple) pour trouver les m états les plus probables du système.

3.2.2 Pour chaque état du système numéroté dans 3.2.1, calculer la demande non acheminée.

3.2.3 Une fois que le flot non acheminé est connu pour chaque état k numéroté dans 2.1 on calcule l'espérance du flot non acheminé au moyen des formules 2.6.3.

Etudions maintenant, en particulier, chacun des points ci-dessus.

3.2.1 Enumération des m états les plus probables du système

Pour faciliter la tâche au lecteur, nous allons reprendre ici les formules developpées par Li et Silvester, et déjà mentionnées dans la section 2.6.

Les pannes des noeuds étant très rares dans un réseau de télécommunications, nous allons supposer que seuls les arcs du système peuvent faire défaillance. Dans le cas où les noeuds sont aussi défaillants il suffit de considérer les noeuds et les arcs comme des "éléments" du système et, au moment de l'étude du réseau défaillant, modéliser la panne d'un noeud comme la panne de tous les arcs qui lui sont incidents.

La deuxième hypothèse à considérer est celle de l'indépendance des pannes. Avant d'étudier l'hypothèse d'indépendance de panne, nous jugeons nécessaire de mentionner l'existence de deux niveaux de réseau: physique et logique. Le réseau physique est celui qui décrit de façon explicite les connexions physiques entre commutateurs. Dans le réseau logique les noeuds représentent les centrales de commutation et les arcs sont constitués par des liens logiques. Il existe un lien logique entre deux centrales s'il y a au moins un chemin physique de connexion entre les centrales en question sans que pourtant il y ait de commutation intermédiaire.

L'hypothèse d'indépendance est donc erronée: dans un réseau logique un lien peut être constitué par un câble qui accueille physiquement plusieurs lignes n'appartenant nécessairement pas à la même connexion logique. Il s'ensuit que la panne d'un lien physique peut entraîner des pannes de plusieurs liens logiques. L'hypothèse d'indépendance des pannes dans le réseau logique n'est donc pas vérifiée. Nous allons contourner cette difficulté et accepter l'hypothèse d'indépendance en travaillant sur le réseau physique au lieu du réseau logique.

Supposons, maintenant, que l'état du réseau est défini par l'état (panne ou fonctionnement) de tous ses arcs. Si n est le nombre d'arcs, le système aura 2^n états possibles.

Posons:

$S_k = k^{ième}$ état du système,

$P(S_k) =$ probabilité que le système soit dans l'état S_k,

$n =$ nombre d'éléments défaillants du système,

$p_i =$ probabilité de bon fonctionnement,

$q_i = 1 - p_i =$ probabilité de défaillance du $i^{ième}$ composante,

alors:

$$P(S_k) = \prod_{i=1}^{n} p_i \left(\frac{q_i}{p_i}\right)^{T_i(S_k)} \tag{2.6.1}$$

où

$$T_i(S_k) = \begin{cases} 0 & \text{si la composante } i \text{ fonctionne} \\ & \text{dans l'état } S_k, \\ 1 & \text{sinon.} \end{cases}$$

La troisième hypothèse considère que la probabilité q_i de défaillance des arcs est toujours inférieure à 50%. Cette hypothèse est nécessaire à l'ordonnancement des états et elle est presque toujours respectée dans la réalité. Posons $R_i = q_i/p_i$, alors les arcs du système sont re-numérotés par ordre croissant de R ($R_1 \geq R_2 \geq R_3 \geq \cdots \geq R_n$) et (2.6.1) devient:

$$P(S_k) = \prod_{i=1}^{n} p_i \, R_i^{T_i(S_k)}. \tag{2.6.2}$$

L'algorithme de Li et Silvester utilise la notion suivante: si $P(S_a) \geq P(S_b) \geq \cdots \geq P(S_c)$ et que $A = \{S_a, S_b, \ldots, S_c\}$ est un ensemble ordonné, alors $B = \{S_a \cup \{i\}, S_b \cup \{i\}, \ldots, S_c \cup \{i\}\}$ sera aussi ordonné. Notons que $S_k \cup \{i\}$ est un nouvel état dont la seule différence avec S_k est la panne de la composante i.

Nous mentionnons, encore une fois, que si les probabilité de pannes des arcs du réseau à étudier sont assez petites, il ne sera pas nécessaire d'utiliser un algorithme d'énumération partielle, car il suffira d'énumerer les états de panne simple du système. Pour illustrer cette situation, nous supposons que tous les arcs du réseau ont une même probabilité de disponibilité p et nous référons le lecteur à la figure 3.1.

La figure mentionnée montre la proportion de l'espace d'états couvert par les états de fonctionnement et de panne simple pour des réseaux de 60, 80 et 100 arcs en fonction de la fiabilité des arcs. La figure 3.2 montre en détail le comportement des courbes de la figure 3.1 lorsque la fiabilité des arcs est supérieure à 99%, ce qui est dans l'ordre de grandeur des disponibilités des arcs d'un réseau de télécommunications.

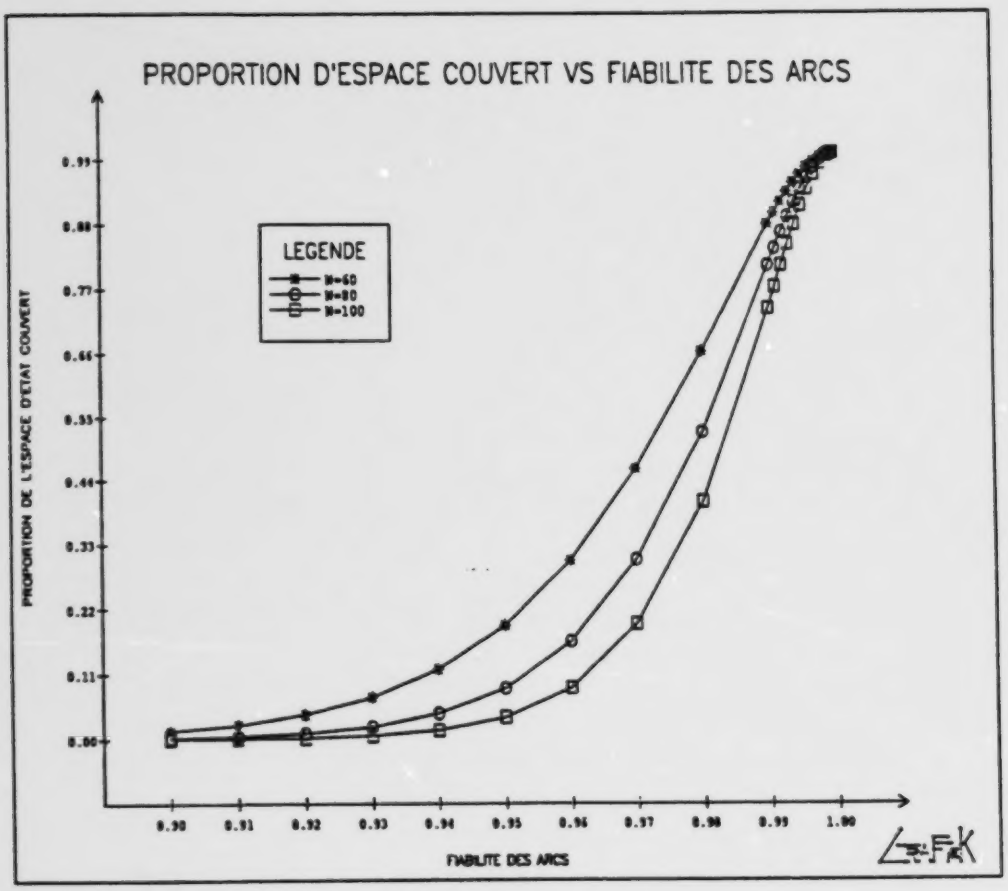


Figure 3.1 Proportion de l'espace d'états couverts par les états de fonctionnement et de pannes simples.

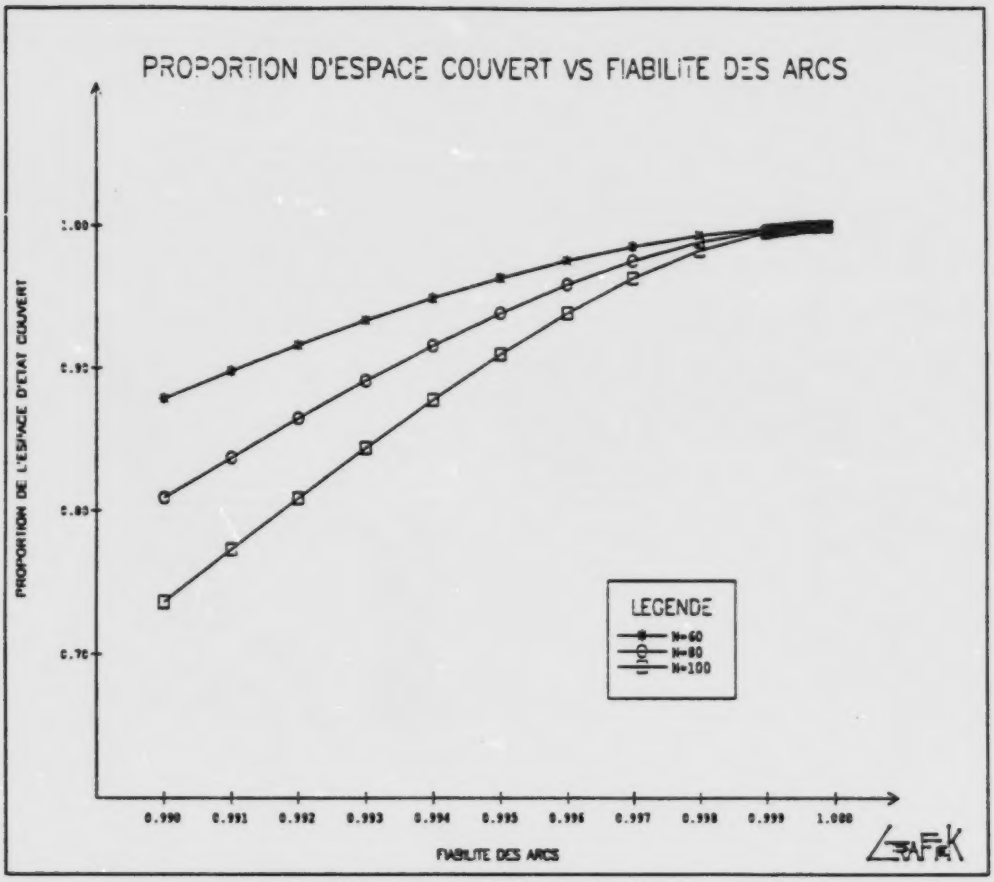


Figure 3.2 Vue en détail de la figure 3.1

Nous verrons, dans le chapitre des résultats, que les probabilités de défaillance des arcs du réseau à étudier sont inférieures à 1%. Il est donc justifiable, pour un réseau d'environ 60 arcs, d'énumerer seulement les états de fonctionnement et de pannes simples, car avec un effort algorithmique négligeable on couvrirait plus de 87% de l'espace total d'états. La figure 3.2 montre que ce chiffre se rapproche dramatiquement de 100% lorsque les fiabilités des arcs dépassent 99.5%.

3.2.2 Calcul de la demande non acheminée pour chacun des m états choisis du réseau

Dans la section précédente, nous avons montré comment trouver sans énumération complète les m états les plus probables du réseau. Pour chaque état, le problème de trouver la demande non acheminée devient un problème d'acheminement optimal d'appels. C'est ainsi que notre objectif principal qui était de définir et de calculer une mesure de fiabilité dans un réseau de télécommunications sera atteint seulement si le sous-problème du routage optimal est résolu efficacement. Ce sous-problème étant un gros problème en soi, nous allons consacrer tout un chapitre de ce travail pour expliquer en détail les modèles de routage proposés. Le premier modèle est un multiflot compatible avec demande déterministe. Dans un deuxième temps, des modèles qui prennent en considération le caractère stochastique de la demande, les problèmes de congestion ainsi que la possibilité d'avoir une hiérarchie dans le réseau sont aussi développés.

Il est à noter que le fait d'incorporer les modèles de routage optimal au calcul de la fiabilité permet à la compagnie de télécommunications d'évaluer la fiabilité sous plusieurs politiques de routage, qu'elles soient implantées ou non.

3.2.3 Calcul des bornes sur l'espérance du flot non acheminé dans le réseau

Une fois que le modèle de routage sera choisi parmi ceux proposés dans le prochain chapitre, on connaîtra la valeur $Z(S_k)$ (flot pondéré non acheminé dans le réseau) pour chaque état S_k du système. Notons que l'on parle de flot pondéré puisque nos modèles permettront de pondérer la demande perdue, en fonction de son origine et sa destination.

Une fois $Z(S_k)$ connu, les bornes sur l'espérance de Z, le flot non acheminé dans le réseau, sont trouvées en considérant uniquement les m états les plus probables et l'état S_{2^n} de défaillance totale:

$$E[Z]_{sup} = \sum_{k=1}^{m} P(S_k)\, Z(S_k) + \left(1 - \sum_{k=1}^{m} P(S_k)\right) Z(S_{2^n})$$

$$E[Z]_{inf} = \sum_{k=1}^{m} P(S_k)\, Z(S_k) + \left(1 - \sum_{k=1}^{m} P(S_k)\right) Z(S_1). \qquad (2.6.3)$$

NOTE: Les équations précédentes ont été numérotées (2.6.3) en dépit du fait que la notation introduite au chapitre 2 a changée. Dans le chapitre 2, on parlait d'un paramètre quelconque à mesurer X. Ici on sait que le paramètre à mesurer est le flot non acheminé dans le réseau. Nous utilisons la variable Z puisqu'elle correspond aussi à la variable qui définit la fonction objectif dans les modèles d'optimisation proposés.

Dans 2.6.3, $Z(S_1)$ est la demande perdue quand il n'y a pas de pannes dans le système, $Z(S_1)$ équivaut donc à la valeur résultante du point 3.1. De façon analogue, $Z(S_{2^n})$ sera la demande perdue lorsque tous les arcs sont défaillants, notons que $Z(S_{2^n})$ doit égaler la demande pondérée.

4. MODÈLES DE ROUTAGE

Dans les chapitres précédents, nous avons fait une révision des mesures de fiabilité des réseaux de télécommunications ainsi que des méthodes couramment utilisées pour évaluer ces mesures. Nous avons aussi exposé une méthodologie pour évaluer la fiabilité du réseau à l'aide des modèles de routage. Jusqu'à ce point, le réseau de télécommunications a été considéré dans sa forme la plus générale. Dans ce chapitre nous définirons de façon plus restrictive le réseau qui nous intéresse sans toutefois entrer dans les détails très techniques de fonctionnement. Les idées essentielles sur le fonctionnement de réseau de télécommunications ainsi que les modèles actuels de routage ont été extraits de Girard (1988).

La distinction entre types de réseaux de télécommunications est basée sur la méthode de commutations d'appels employée. Un appel est ici considéré dans le sens le moins restrictif; il peut représenter une conversation téléphonique aussi bien qu'un échange de données avec un ordinateur. Nous verrons, cependant, que certains types de commutation sont plus appropriés à certains types d'appels.

Sans entrer dans les détails du fonctionnement du réseau, disons de façon très simpliste qu'une demande d'appel met en marche une procédure à trois étapes. D'abord le contrôle d'accès d'appel détermine si ce dernier doit être accepté ou non. Si l'appel est accepté, une affectation des ressources nécessaires à l'acheminement de l'appel sera réalisée. La troisième étape, le contrôle du flot, assure le respect de certains paramètres.

C'est l'étape d'allocation des ressources qui caractérise le type de réseau de télécommunications. Si toutes les ressources nécessaires à l'acheminement de l'appel sont réservées pendant la durée complète de la connexion, on parle de commutation par circuits. Si, par contre, les ressources sont réservées de façon partielle, seulement pour la durée de la transmission de l'information entre commutateurs, l'on parle de commutation par paquets.

Lorsque les appels occupent la ligne de transmission pendant une bonne partie du temps de connexion, la commutation par circuits est avantageuse. C'est le cas de la transmission de la voix lors des appels téléphoniques. Si le fac-

teur d'utilisation est bas, comme dans le cas des communications par ordinateur, l'information à transmettre peut être découpée en unités d'information et envoyée uniquement lorsque chaque unité est complète. Dans ce cas, une allocation totale des ressources constituerait un véritable gaspillage et on utilise alors la commutation par paquets.

Une autre caractéristique qui détermine le choix du type de réseau à utiliser est l'erreur ou le délai accepté par les parties impliquées dans l'appel. L'utilisateur d'un terminal d'ordinateur accepte un certain délai dans la transmission de l'information, par contre, il n'aime certainement pas recevoir des données erronnées. De façon inverse, les utilisateurs du téléphone acceptent que la clarté de la voix ne soit pas parfaite, mais un long délai dans la transmission les empêcherait de se bien comprendre. La commutation par paquets admet un certain délai qui n'est pas présent dans la commutation par circuits. Cependant des codes de vérification garantissent que les erreurs dans la transmission soient minimales. D'un autre côté, les appels dans un réseau à commutation par circuits sont acceptés ou refusés tandis que le réseau à commutation par paquets accepte presque toutes les demandes d'appels créant, parfois, une file d'attente pour leur acheminement.

Il s'ensuit que la nature des modèles mathématiques pouvant optimiser ou simplement étudier le routage d'appels dans les deux types de réseaux est très différente. Les modèles développés dans ce chapitre sont propres au réseau à commutation par circuits.

Une particularité supplémentaire des réseaux de télécommunications par rapport à d'autres réseaux de flots est le fait qu'il n'est pas facile d'identifier la nature du flot à transporter. En effet, cela n'a pas beaucoup de sens de parler de "flot d'appels" comme on parlerait de mégawatts dans des réseaux électriques ou de voitures dans ceux de transport, les appels étant des entités physiques avec une certaine durée. Pour bien saisir la notion du flot dans le réseau de télécommunications, nous allons lui consacrer la première section du chapitre. Une revue des types de routage existants est faite dans la section deux, le reste de ce chapitre étant réservé aux modèles de routage proposés.

4.1 Nature du flot dans un réseau de télécommunications:

Pour faire l'étude du routage d'appels dans le réseau de télécmmunications, nous nous placerons dans une période de temps T et on y étudiera le flot de trafic. Dans Bear (1980) plusieurs définitions sont données au flot de trafic dans un câble à plusieurs lignes.

Soit:

T =longueur de la période en étude,

volume de trafic =somme sur la période T des durées des appels passant par le câble,

C =moyenne du nombre d'appels arrivant au système par unité de temps,

h =durée moyenne d'un appel,

nous avons alors

Définition 1: flot de trafic $=$ $\dfrac{\text{Volume de trafic}}{T}$.

Définition 2: flot de trafic $=$ Moyenne du nombre de lignes occupées simultanément pendant la période T.

Définition 3: flot de trafic $=$ $C \bullet h$.

Les trois définitions sont équivalentes et ont la même unité de mesure: l'Erlang. Si une seule ligne est occupée de façon continue ou non pendant un temps t à l'intérieur de la période T, on dit qu'elle a un trafic de t/T Erlangs et son trafic maximal est 1 Erlang. 1 Erlang et ainsi équivalent à un circuit occupé à plein temps. L'Erlang est donc une unité sans dimension.

Passons maintenant à la notion de demande dans les réseaux qui nous intéressent. Par définition, la demande représente le nombre moyen d'essais effectués dans une période égale à la durée moyenne d'un appel. Dans la terminologie propre aux ingénieurs en télécommunications la demande est appelée "trafic

offert" par opposition au flot accepté qui est appelé "trafic écoulé". Le flot refusé est appelé "flot de débordement".

Les termes "demande" ainsi que "trafic accepté" ou "refusé" appartiennent à la terminologie de réseau, il s'ensuit que, tout en espérant que cela ne créera pas de confusion, les deux terminologies seront utilisées dans le contexte qui nous semblera le plus approprié.

4.2 Techniques de routage

Nous allons, dans cette section, définir et faire une petite révision des différentes techniques de routage existantes. Les définitions présentées ont été extraites de Girard.

Le routage constitue le processus du choix d'une route pour acheminer un appel. Il existe plusieurs types de routage. Si l'aspect temporel est pris en considération, soit la demande soit certains paramètres varient avec le temps. À cette classe appartiennent le routage dynamique et le routage adaptatif. Le terme "dynamique" indique simplement qu'une partie du routage varie avec le temps. Le routage adaptatif est celui où une partie du routage est fonction de certains paramètres donnés dans le réseau. Ce dernier routage est généralement dynamique, malgré que l'on puisse avoir un acheminement adaptif même si le processus d'arrivée est stationnaire, tandis qu'un routage dynamique n'est pas nécessairement adaptatif.

Les modèles de routage développés dans ce travail considèrent que la demande ne varie pas pendant la période d'étude. Ils appartiennent alors au type de routage fixe.

Il existe d'autres classifications du routage reliées aux règles de commutation plutôt qu'à l'aspect temporel de l'étude du réseau. Pour bien saisir les distinctions entre les concepts à donner, revisons les étapes qui traditionnellement ont caractérisé le routage. D'abord il s'agit de déterminer tous les chemins possibles à l'acheminement d'un appel. La deuxième étape est constituée par le choix

du chemin à utiliser; ce qui nécessite une règle de commutation. La commutation par paquets présente une étape additionelle du routage qui détermine si un appel sera ou non acheminé même s'il y a un chemin disponible. La dernière étape est constituée par la gestion des appels qui ne trouvent pas de chemin disponible.

C'est au niveau de la deuxième étape que la classification du routage est faite. Un type de règle populaire de routage est appelé routage avec débordement fixe: un appel bloqué sur un chemin est offert au suivant et ainsi de suite jusqu'à la fin. Parmis les classes les plus importantes de routage fixes alternatifs l'on trouve le routage à deux liens et routage hiérarchique. Le premier consiste à assigner seulement des chemins à deux liens comme alternative pour l'acheminement des appels entre chaque pair de noeuds origine-destination. Pour introduire la définition du routage hiérarchique, il faut mentionner que les commutateurs du réseau sont classifiés en suivant une hiérarchie pré-établie. Le routage hiérarchique est alors caractérisé par la nature des chemins alternatifs, chemins dont les noeuds ou commutateurs appartiennent à des classes différentes.

Un autre type de règle de routage est le routage sans débordement avec partage de charge. Il s'agit de repartir la demande d'une paire O/D sur plusieurs routes pré-établies d'après un coefficient de proportionalité attribué à chacune des routes. Ces coefficients sont généralement calculés à partir de la probabilité de blocage des routes. Ce type de routage est appellé "sans débordement" parce que les appels qui ne réussissent pas à passer sont, tout simplement, perdus. Cette règle de routage est beaucoup plus proche d'un problème de flot que le routage avec débordement fixe. En effet, les modèles de routage avec débordement fixe ne constituent qu'une règle qui décrit comment acheminer un appel une fois qu'il se présente sans qu'on ait aucun control sur ce qui se passe dans le système. Par contre les modèles de partage de charge permettent une plus grande souplesse si l'on veut optimiser le routage, on peut, par exemple, vouloir changer les coefficients de proportionnalité pour réduire le blocage. Bien que le routage avec partage de charge n'est pas utilisé en pratique, le routage avec débordement étant moins coûteux à mettre en place, il constitue une approximation nécessaire par la difficulté de l'optimisation de l'acheminement. De plus, les modèles avec partage de charge peuvent devenir adaptifs. Ils gagnent donc de plus en plus d'importance.

La classification du routage que l'on vient de voir est inhérente à l'utilisation d'une technologie "classique" d'acheminement d'appels.

Les modèles que l'on montrera dans notre étude ont été crées en tenant compte des nouvelles technologies qui permettent beaucoup plus de souplesse à la procédure de routage. Ils intègrent les trois étapes qui caractérisent les routages dans une seule et optimisent la gestion des appels dans le réseau. Puisqu'ils ont été conçus en vue de l'optimisation de la gestion du routage d'appels, nous allons classifier nos modèles d'après le type de politique d'optimisation utilisée.

4.3 Modèles de routage dans un réseau de télécommunications:

Nous allons avant tout faire une révision de certains modèles de routage utilisant l'idée d'optimisation globale du réseau de télécommunications est présente. Il s'agit des modèles de Kortanek et Polak (1984) développés dans un contexte de conception de réseaux. Leur problème est de déterminer les capacités optimales d'un réseau non hiérarchique avec des demandes de flot qui varient avec le temps. Le problème est donc ramené à un problème de routage dynamique.

DDR *(Modèle de routage déterministe)*. Il s'agit d'un modèle déterministe qui minimise les coûts des capacités des lignes tout en respectant la demande de chaque paire O/D pour chaque heure. La formulation est du type multiflot compatible (avec contrainte de capacité) sous forme arcs-chemin.

SDR *(Modèle de routage dynamique)*. À la différence du DDR, le SDR prend en considération le fait que les flots sont des variables aléatoires distribuées sous forme d'Erlang. La formulation sous forme arc-chemin est encore un multiflot avec contrainte de capacité mais avec une fonction de pénalité dans l'objectif qui reflète le caractère stochastique de la demande. Dans l'article étudié il n'est pas clair comment cette fonction a été construite. De plus, nous verrons un peu plus loin dans nos modèles qu'il n'est pas nécessaire de garder la contrainte de capacité si l'on utilise une bonne fonction de pénalité dans l'objectif.

4.3.1 Modèles développés d'optimisation centralisée:

Dans cette section, nous allons proposer des modèles qui obéissent à une politique d'optimisation globale ou "centralisée". Il existe un autre type d'optimisation appelée "égoïste" ou "décentralisée". La discussion sur ce qui différencie les deux notions est rapportée à la section 4.3.2.

4.3.1.1. Modèles déterministes:

MD1: (Premier modèle déterministe). L'idée de ce premier modèle est de trouver l'acheminement optimal des flots O/D étant donné que l'on veut minimiser la demande pondérée par O/D qui ne peut pas être acheminée dans le réseau. Le modèle proposé est un multiflot avec contrainte de capacité. Les différents flots ou produits à considerer sont constitués par les flots de trafic d'appels allant d'un noeud origine à une noeud destination. Il y a autant de produits que de paires de noeuds O/D. Le modèle proposé présente une formulation par arcs et l'unité de flot utilisée est l'Erlang.

Soit:

X_u^{pq} =flots d'origine p et destination q, traversant l'arc u,

C_u =capacité de l'arc u,

d^{pq} =demande d'origine p et de destination q,

Y^{pq} =demande d'origine p et de destination q ne pouvant pas être acheminée,

$\mathcal{A}^-(i)$=ensemble des arcs se terminant au noeud i,

$\mathcal{A}(i)$ =ensemble des arcs débutant au noeud i,

k^{pq} =pondération attribuée au flot d'origine p et de destination q,

O =ensemble des noeuds origine,

D =ensemble des noeuds destination,

N =ensemble des noeuds du graphe,

M =ensemble des arcs du graphe,

n =cardinalité de N,

m =cardinalité de M.

MD1: (Premier modèle déterministe).

$$\min Z = \sum_{p \in O} \sum_{q \in D} k^{pq} Y^{pq}$$

s. à

Capacité :

$$\sum_{p \in O} \sum_{q \in D} X_u^{pq} \leq C_u \qquad \forall u \in M.$$

Conservation de flot :

$$\forall p \in O,\ \forall q \in D,\ \forall i \in N$$

$$\sum_{u \in \mathcal{A}(i)} X_u^{pq} - \sum_{u \in \mathcal{A}^-(i)} X_u^{pq} = \begin{cases} d^{pq} - Y_{pq} & \text{si} \quad i = p, \\ -d^{pq} + Y^{pq} & \text{si} \quad i = q, \\ 0 & \text{sinon.} \end{cases}$$

Nonnégativité :

$$\forall p \in O, \forall q \in D, \forall u \in M$$

$$X_u^{pq} \geq 0, \quad Y^{pq} \geq 0.$$

Le nombre de contraintes du problème tel que posé est $\mathcal{O}(n^3)$ et le nombre de variables, $\mathcal{O}(n^4)$. Reformulons le problème en définissant un produit comme le flot qui sort d'un même noeud origine. Ainsi le nombre de contraintes est réduit à $\mathcal{O}(n^2)$ et les variables à $\mathcal{O}(n^3)$. Cette formulation donne lieu au deuxième modèle déterministe MD2.

Soit:

X_u^p =flot sortant du noeud p qui traverse l'arc u,

D_p =ensemble des noeuds destination des appels originant au noeud p.

MD2: (Deuxième modèle déterministe).

$$\min Z = \sum_{p \in O} \sum_{q \in D} k^{pq} Y^{pq}$$

s. à

Capacité :

$$\sum_{p \in O} X_u^p \leq C_u \qquad \forall u \in M.$$

Conservation de flot :

$$\forall p \in O, \forall i \in N$$

$$\sum_{u \in \mathcal{A}(i)} X_u^p - \sum_{u \in \mathcal{A}^-(i)} X_u^p = \begin{cases} \sum_{q \in D_p} \left(d^{pq} - Y^{pq} \right) & \text{si} \quad i = p, \\ -d^{pi} + Y^{pi} & \text{si} \quad i \in D_p, \\ 0 & \text{sinon.} \end{cases}$$

Nonnégativité :

$$\forall p \in O\,,\ \forall u \in M\,,\ \forall q \in D$$

$$X_u^{pq} \geq 0\,,\ Y^{pq} \geq 0.$$

Les modèles proposés considèrent que le trafic pour une paire origine-destination peut utiliser n'importe quel chemin du réseau. Il est possible de modifier la fonction objectif afin de favoriser le plus court chemin ou pour éviter le problème du délestage dont on parlera plus tard.

Modification des fonctions objectifs des modèles MD1 et MD2

Modification pour favoriser le plus court chemin

Pour MD1, si l'on favorise le plus court chemin, nous avons l'objectif suivant:

$$\min Z_1$$

$$Z_1 = \sum_{p \in O} \sum_{q \in D} k^{pq} Y^{pq} + \sum_{u \in M} \sum_{p \in O} \sum_{q \in D} \ell_u w_u X_u^{pq}$$

où

ℓ_u = longueur de l'arc u,

w_u = pondération donnée à l'arc u.

Pour MD2 on a:

$$\min Z_2$$

$$Z_2 = \sum_{p \in O} \sum_{q \in D} k^{pq} Y^{pq} + \sum_{u \in M} \sum_{p \in O} \ell_u w_u X_u^{p}.$$

Modification pour empêcher le délestage:

Suite à une panne dans le réseau, il sera nécessaire de réaliser une ré-optimisation du routage. La présence des longueurs des arcs dans la fonction d'optimisation réduit un long détour mais il sera parfois souhaitable d'éviter certains chemins. En effet, si une panne se produit, on veut que le flot soit ré-acheminé par le plus court chemin, mais que, autant que possible, la panne n'affecte que les noeuds entre lesquels elle s'est produite. Ceci n'est pas toujours le cas; si, par exemple, une panne se produit sur un arc reliant deux clients importants, le poids de ces clients sur la fonction objectif rendra leur demande de lignes prioritaire. Il s'ensuit que deux clients d'importance inférieure qui sont localisés loin de la panne pourraient se retrouver avec leurs lignes bloquées par des appels ré-acheminés.

Nous envisageons deux stratégies pour éviter le problème décrit ci-dessus (problème du délestage). Il s'agit de pénaliser ou d'interdire le passage d'un certain flot sur un certain arc.

Pour la première formulation (MD1) posons:

$I =$ ensemble d'arcs où il n'est as souhaitable d'acheminer le flot d'origine p et destination q.

Dans ce cas la nouvelle fonction objectif Z_3 aura la forme de Z_1 plus un terme de pénalité P,

$$P = \sum_{p \in O} \sum_{q \in D} \sum_{u \in I} M_u^{pq} X_u^{pq}$$

où M_u^{pq} est un grand poids attribué à la pénalité de passage du flot d'origine p et destination q sur l'arc u.

L'autre possibilité pour éviter le délestage est l'interdiction du passage de certains flots sur certains chemins. La fonction objectif reste Z_1 mais on ajoute à la formulation les contraintes suivantes:

$$X_u^{pq} = 0 \qquad \forall p \in O,$$
$$\forall q \in D,$$
$$\forall u \in I.$$

En pratique il suffit de supprimer les variables pour que les contraintes soient respectées.

Nous venons de voir les modifications à faire pour la première formulation (celle où les produits sont les flots de trafic d'appels entre paires origine-destination). Ces mêmes modifications sont possibles pour la deuxième formulation (MD2), mais les pénalités ou les interdictions pourraient apporter des restrictions inutiles au routage. Posons:

J = ensemble des arcs où il n'est pas souhaitable d'acheminer le flot originant de p.

Alors Z_4, la nouvelle fonction objectif, a la forme de Z_2 plus le terme P de pénalité:

$$P = \sum_{p \in O} \sum_{u \in J} M_u^p X_u^p$$

où M_u^p est un grand poids de pénalité donné au passage du produit p sur l'arc u

Les contraintes d'interdiction seraient par contre:

$$X_u^p = 0 \qquad \forall p \in O,$$
$$\forall u \in J.$$

Le problème d'utiliser les pénalités ou les interdictions avec MD2 pourrait se présenter lorsqu'on interdit le passage d'un flot sur une ligne où il circulerait de façon naturelle. La figure 4.1 montre un exemple:

Supposons que l'on ne veut pas qu'une panne de l'arc (a, e) affecte l'arc (w, y). Si l'on travaille avec MD1, on a simplement à poser les pénalités ou les interdictions sur le flot $X_{(w,y)}^{ae}$.

Pour MD2, on devrait imposer des restrictions sur le flot du type a qui passe par (w, y), $X^a_{(w,y)}$. Ceci empêcherait que la panne de (a, e) ait une influence sur (w, y). Mais le produit de type a inclut tous les appels qui se font à partir du noeud a. La restriction sur $X^a_{(w,y)}$ affecte alors le flot naturel entre a et b. Il s'ensuit que les pénalités ou interdictions possibles avec la deuxième formulation sont plus limitées et doivent être choisies soigneusement par la compagnie de communications.

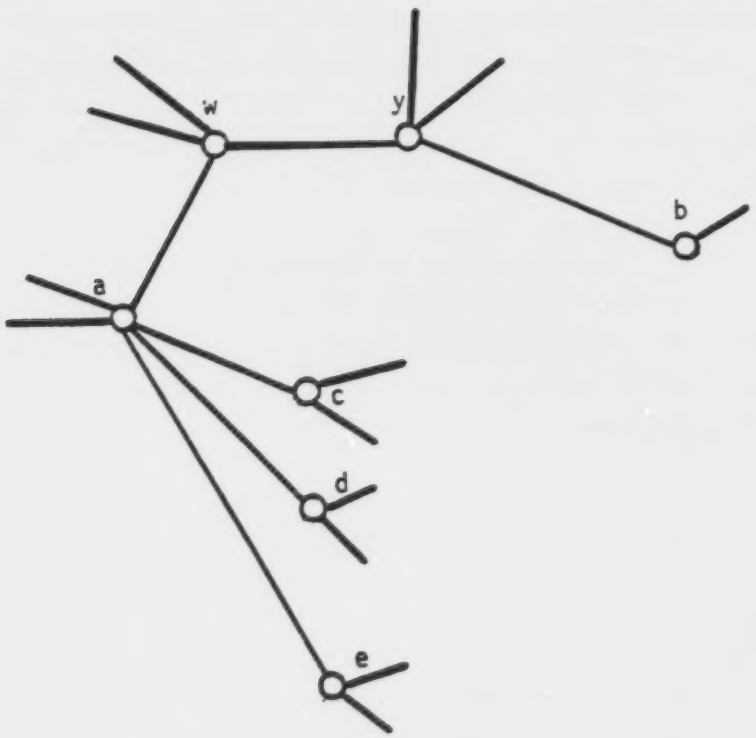


Figure 4.1 Exemple qui montre le problème d'interdire certains flots si l'on utilise MD2.

Le choix de la modélisation dépend alors des besoins du réseau. S'il s'agit d'un petit réseau, la première formulation conviendrait mieux puisque le prix de

$\mathcal{O}(n)$ à payer pour pouvoir avoir de la flexibilité sur les restrictions de délestage vaut le coût. Par contre, le problème de ré-acheminement et délestage se présente plus probablement sur des grands réseaux où un facteur de $\mathcal{O}(n)$ sur la taille du réseau peut être très significatif.

4.3.1.2. Modèles stochastiques

Les modèles déterministes MD1 et MD2 sont extrêmement difficiles à résoudre à cause de la contrainte de capacité qui lie les flots de différents produits entre eux. Traditionnellement, une des façons de résoudre ce type de formulation était de remplacer la contrainte de capacité par une fonction de pénalité dans l'objectif. En général la fonction de pénalité était construite de façon "artificielle" à partir de la contrainte éliminée. Le mot "artificielle" signifie que malgré l'utilité de la pénalité pour éviter le débordement, elle n'est pas un facteur qu'on souhaiterait minimiser.

L'idée des modèles stochastiques est d'éliminer la contrainte de capacité et d'ajouter, à la place, une pénalité qui reflète le caractère stochastique de la demande tout en empêchant que la capacité des arcs soit dépassée. Cette idée à déjà été mise en pratique avec succès dans des modèles de transport aérien (voir Soumis (1979) et Lavoie (1984)).

L'approche est expliquée à l'aide de la figure 4.2. Cette figure représente la distribution probabiliste du flot acheminé sur l'arc u. La région rayée dans la figure équivaut à la probabilité que le flot sur l'arc dépasse sa capacité.

Nous avons choisi de travailler avec des espérances au lieu de distributions probabilistes. Introduisons une fonction d'espérance dont la minimisation pénalise la capacité dépassée sur les arcs. Prenons, par exemple, l'espérance du flot non acheminé sur un arc. Idéalement on voudrait trouver une fonction qui se comporte comme celle représentée dans la figure 4.3.

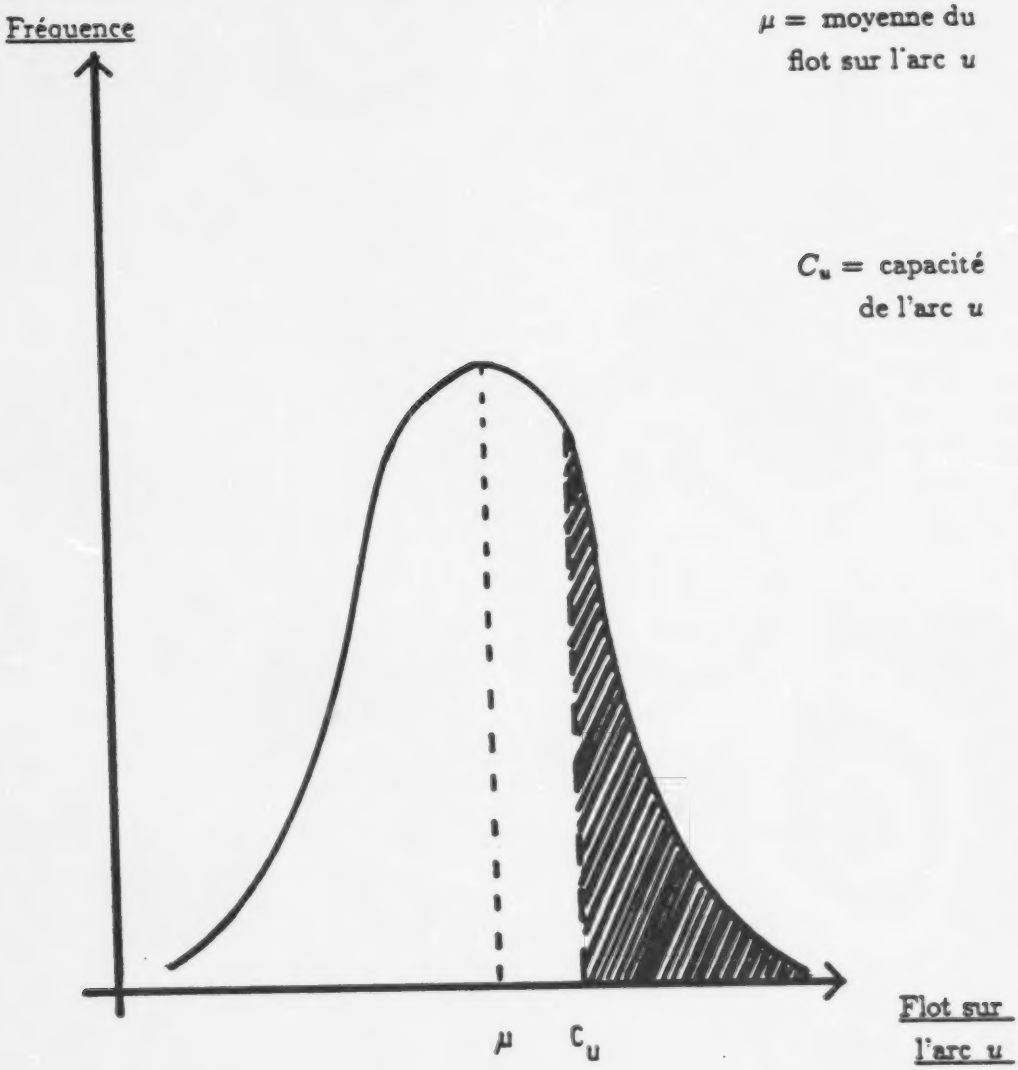


Figure 4.2 Distribution de la probabilité du flot acheminé sur l'arc u.

NOTE : Pour des raisons de commodité, 4.2 représente la distribution du flot comme étant une cloche; ceci n'est pas nécessairement le cas dans la réalité.

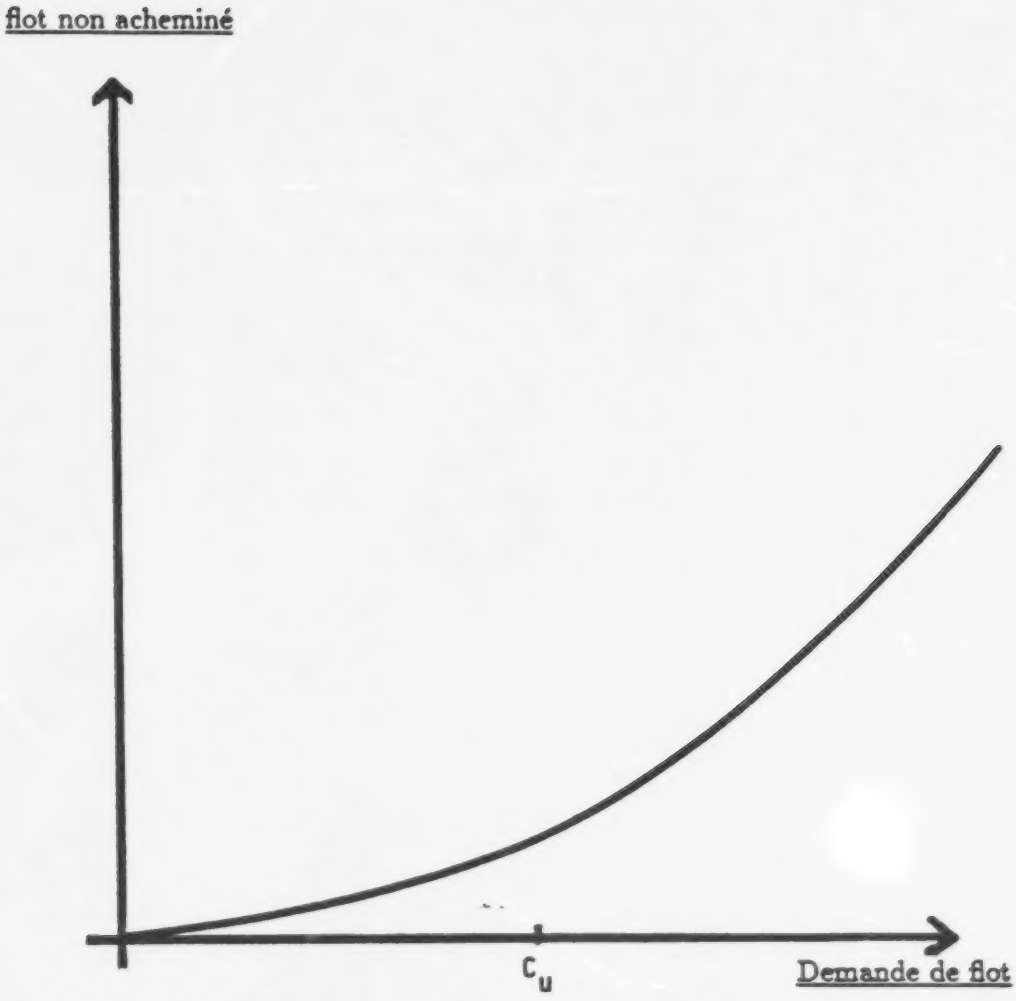


Figure 4.3 Espérance du flot non acheminé sur l'arc u.

Nous avons maintenant deux options: construire une fonction de l'espérance du flot refusé en nous basant sur des données réelles fournies par la compagnie de télécommunications ou faire des hypothèses à propos de la distribution des appels refusés. Le développement pour arriver à la probabilité de refus d'une unité d'appel a été fondé par A.K. Erlang en 1918 (voir Bear (1980)). Pour arriver à sa fameuse "formule de pertes", Erlang a fait les hypothèses suivantes:

Hypothèse 1: Les appels ont lieu de façon aléatoire et leur arrivée dans le temps peut être décrite par un processus de Poisson.

Hypothèse 2: On est dans un état d'équilibre statistique.

Hypothèse 3: Les appels demandés lorsque toutes les lignes disponibles sont occupées sont tout simplement perdus, il n'y a pas de file d'attente. Les tentatives subséquentes de placer un appel sont aussi négligées.

Hypothèse 4: La distribution des temps de durée des appels est exponentielle. Cette dernière hypothèse s'est avérée redondante en pratique et, en 1948, Kosten a démontré qu'elle n'était pas nécessaire.

Nous n'allons pas démontrer la formule de pertes d'Erlang, deux démonstrations possibles sont clairement montrées dans Bear. La formule correspond à la proportion d'appels perdus ou à la probabilité de trouver toutes les lignes occupées $B(s, a)$:

$$B(s, a) = \frac{a^s / s!}{\sum_{k=0}^{s} a^k / k!} \tag{4.1}$$

où :

a = trafic offert (en unités de flot),

s = capacité de l'arc (en nombre de lignes).

A partir de la formule (4.1) nous pouvons déduire l'espérance du trafic perdu:

$$E\left(Z(C_u, a)\right) = a\, B(C_u, a) \tag{4.2}$$

où

Z = trafic perdu sur l'arc u,

C_u = capacité de l'arc u.

Nous avons maintenant tous les éléments pour introduire le premier modèle stochastique.

MS1 (Premier modèle stochastique)

Ce modèle considère que les produits sont équivalents aux flots origine-destination. Pour ce qui est de la fonction de pénalité, selon l'équation 4.2, si X_u^{pq} est le flot d'origine p et destination q qui essaie de passer par l'arc u alors la pénalité à ajouter à la fonction objectif sera:

$$P(X) = \sum_{u \in M} \sum_{p \in O} \sum_{q \in D} X_u^{pq} B\left(C_u, \sum_{p \in O} \sum_{q \in D} X_u^{pq}\right) \tag{4.3}$$

La fonction objectif Z_1 est alors composée d'un terme de pénalité et d'un terme de choix du chemin le plus convenable. De plus, Z_1 est convexe car P est convexe. La convexité de P est étudiée à la fin du chapitre. Le modèle MS1 est formulé de la façon suivante:

MS1 :

$$\min Z_1 = \sum_{u \in M} \sum_{p \in O} \sum_{q \in D} \ell_u w_u X_u^{pq} + P(X)$$

s. à

Conservation de flot :

$$\forall p \in O, \forall q \in D, \forall i \in N$$

$$\sum_{u \in A(i)} X_u^{pq} - \sum_{u \in A^-(i)} X_u^{pq} = \begin{cases} d^{pq} & \text{si} \quad i = p, \\ -d^{pq} & \text{si} \quad i = q, \\ 0 & \text{sinon.} \end{cases}$$

Nonnégativité :

$$X_u^{pq} \geq 0 \quad \forall u \in M, \quad \forall p \in O, \quad \forall q \in D.$$

Le terme d^{pq} est la demande moyenne de flot d'appels de p à q.

Cette formulation a l'avantage d'éliminer la contrainte de capacité, ce qui facilite énormément la résolution du problème. De plus, tel que mentionné, la fonction de pénalité porte sur un facteur que l'on souhaite minimiser dans le système. On peut donc s'attendre à ce que le modèle se comporte mieux que si la contrainte avait été créée artificiellement.

Il est à noter que dans MS1, la variable X_u^{pq} ne représente plus le trafic exact allant de p à q à travers l'arc u mais plutôt une intensité de trafic que l'on essaie d'acheminer sur l'arc. Plus précisément, (X_u^{pq}/d^{pq}) représente la proportion des appels de p à q qui seront envoyés sur l'arc u. On doit aussi mentionner que dans notre modèle la probabilité de perdre un appel envoyé sur le chemin $u_1, u_2, \ldots, u_k$ est implicitement considérée comme égale à

$$\sum_{j=1}^{k} B\left(C_{u_j}, \sum_{p\in O}\sum_{q\in D} X_{u_j}^{pq}\right)$$

tandis qu'un estimé plus correct serait:

$$\left[1-\prod_{j=1}^{k}\left(1-B(C_{u_j}, \sum_{p\in O}\sum_{q\in D} X_{u_j}^{pq})\right)\right]$$

(cet estimé est encore erroné car il est basé sur l'hypothèse que les flots transportés sur les arcs à n'importe quel instant constituent des variables aléatoires indépendantes — ce qu'ils ne sont pas). On est donc conscient de l'erreur introduite par l'utilisation de la formule additive de probabilité de pertes dans le réseau. Il est nécessaire d'introduire cette erreur si l'on veut avoir une fonction $P(X)$ séparable par arcs, ce qui facilite la résolution du problème. Il s'ensuit que la valeur de pertes dans le réseau obtenue par l'expression (4.3) constitue une borne supérieure de la vraie valeur du trafic bloqué dans le réseau.

Afin de diminuer le nombre des variables, nous proposons un modèle stochastique similaire à MD2 que nous allons appeler MS2.

MS2 (Deuxième modèle stochastique)

$$\min Z_2 = L(X) + P(X)$$

s. à

Conservation de flot:

$$\forall p \in O,\ \forall u \in M,\ \forall i = 1, \ldots n$$

$$\sum_{u \in \mathcal{A}(i)} X_u^p - \sum_{u \in \mathcal{A}^-(i)} X_u^p = \begin{cases} \sum_{q \in D_p} d^{pq} & \text{si} \quad i = p, \\ -d^{pi}, & \text{si} \quad i \in D_p, \\ 0 & \text{sinon.} \end{cases}$$

Nonnégativité :

$$X_u^p \geq 0 \qquad \forall p \in O \qquad \forall u \in M$$

où,

$$P(X) = \sum_{u \in M} \sum_{p \in O} k^p X_u^p B\left(C_u, \sum_{p \in O} X_u^p\right).$$

$L(X)$ est une fonction qui favorise le plus court chemin et empêche les problèmes de délestage.

4.3.2. Optimisation centralisée versus optimisation décentralisée:

Dans les modèles montrés jusqu'à maintenant, l'objectif est de minimiser de façon globale le flot d'appels excédentaires sur les arcs. Le problème peut toutefois être abordé de façon complètement différente. Il s'agit de considérer les paires origine-destination comme des marchés d'appels et d'optimiser en fonction de

chaque marché en particulier. Ce type d'optimisation est appelée "décentralisée" puisque les formulations des problèmes essaient d'optimiser l'intérêt de chaque marché plutôt que de chercher la satisfaction de l'ensemble du réseau. Cette notion, sous la terminologie d'optimisation "égoïste", est amplement utilisée en transport routier.

L'intérêt que l'on porte à l'optimisation décentralisée plutôt qu'à l'optimisation coopérative ou globale est dû au fait qu'elle est plus proche des règles de routage qui existent actuellement en télécommunications. On a considéré que la fiabilité peut être mesurée en fonction du flot d'appels excédentaires dans le réseau. Cette mesure, dans un modèle d'optimisation coopérative, est très faible, car il y a une espèce de centre de routage extrêmement efficace qui place des appels un peu partout de façon à ce qu'ils soient toujours acheminés. L'optimisation égoïste, par contre, reflète un routage décentralisé; les appels sont acheminés suivant une certaine politique et, si cette dernière n'est pas respectée, les appels seront perdus. La mesure d'appels non acheminés est alors plus forte dans un modèle décentralisé. Il s'ensuit que ce type de modèle évalue de façon plus sévère la performance du réseau.

Avant de passer au développement des modèles d'optimisation égoïste, nous allons faire une petite synthèse de la bibliographie qui touche à ce sujet.

La notion d'optimisation égoïste a été formalisée par Wardrop dans un contexte de transport routier. Son fameux "premier principe" établit que selon la perception de l'usager et sous des conditions actuelles de trafic, les routes dont le temps de parcours est le plus court sont celles où il y a du flot de trafic. C'est aux usagers d'une paire origine-destination de choisir la route qu'ils croient la meilleure même si le temps de parcours du système n'est pas optimisé.

Le deuxième principe de Wardrop veut que "le temps moyen de parcours du système soit minimum". Le premier principe est plus rapproché de la réalité puisqu'il implique qu'à l'optimum aucun usager ne pourra réduire son temps de parcours en choisissant une route alternative, le deuxième principe, par contre, correspond à une "optimisation globale" et globalement donnera un comportement plus efficace du système.

Le modèle d'équilibre formulé, pour que pour chaque paire O/D le premier principe soit satisfait il faut que $\forall\, k \in K_i$ et pour $\forall\, i \in I$,

$$s_k^* \begin{cases} = u_i^* & \text{si} \quad h_k^* > 0 \\ \geq u_i^* & \text{si} \quad h_k^* = 0 \end{cases} \tag{4.6}$$

s. à

$$\sum_{k \in k_i} h_k = g_i, \tag{4.7}$$

$$h_k \geq 0, \tag{4.8}$$

où

I = ensemble des paires O/D,

K_i = ensemble des routes pour la paire $O/D\, i$,

g_i = demande de la paire $O/D\, i$,

h_k = flot sur la route k,

s_k = coût de la route k,

$u_i = \min\limits_{k \in k_i} s_k$.

Les paramètres étoilés dans (4.6) correspondent aux valeurs de l'équilibre.

Ecrivons maintenant la fonction de temps de parcours, c.-à-d., la fonction de coûts, en fonction du flot sur les arcs. Le cheminement pour arriver à l'expression (4.9) est expliqué dans Florian (1982).

$$\sum_{a \in A} s_a(v_a^*)(v_a - v_a^*) \geq \sum_{i \in I} u_i^*(g_i - g_i^*) \tag{4.9}$$

où:

A = ensemble des arcs du réseau,

v_a = flot de trafic sur l'arc a,

d_i = demande de la paire $O/D\, i$.

Il est possible d'inverser le vecteur demande. Soit $W(g) = u$ l'inverse de la fonction de demande, alors le modèle d'équilibre est décrit par:

$$\sum_{a \in A} s_a(v^*)(v_a - v_a^*) - \sum_{i \in I} W_i(g^*)(g_i - g_i^*) \geq 0 \qquad (4.10)$$

s. à (4.7) et (4.8).

L'avantage de la formulation précédente est de donner un modèle en termes d'arcs qui correspond aux conditions de Kuhn-Tucker du problème d'optimisation suivant:

$$\min Z = \sum_{a \in A} \int_0^{v_a} s_a(x)\,dx + \sum_{i \in I} \int_0^{g_i} W_i(y)dy$$

s. à

$$\sum_{k \in K_i} h_k = g_i \qquad \forall i \in I \qquad (4.11)$$

où

$$v_a = \sum_{i \in I} \sum_{k \in K_i} \delta_{ak} \cdot h_k$$

et

$$\delta_{ak} = \begin{cases} 1 & \text{si} \quad a \in \text{ route } k, \\ 0 & \text{sinon.} \end{cases}$$

Dans certains cas, les équations qui décrivent le comportement égoïste d'une paire O/D ont une correspondance avec un modèle d'optimisation dont les algorithmes de résolution sont connus. Il s'ensuit qu'il est parfois souhaitable de transformer les équations qui décrivent le comportement égoïste des usagers à leur équivalent en optimisation du réseau.

4.3.3 Modèles stochastiques décentralisés

Revenons à la notion de routage avec débordement fixe décrite dans la section 4.2. L'ordre dans lequel les routes alternatives seront considérées lors de l'acheminement d'un appel peut être établi dans des tables de routage. Maintenant, les décisions reliées au choix du chemin que l'appel doit emprunter peuvent être prises à partir du noeud origine ou, de façon sequentielle, au fur et à mesure que l'appel avance, au niveau de chaque commutateur.

Lorsqu'on optimise le routage pour l'état stable, les décisions prises de façon séquentielle doivent être équivalentes à celles qui sont prises à l'origine de l'appel. Notons que l'affirmation précédente est vérifiée uniquement s'il existe du "crankback" dans le réseau, c.-à-d. les appels n'ayant pas trouvé de la place sur une première route peuvent revenir à l'origine et essayer sur la deuxième route. Nous allons alors adopter pour nos modèles une politique de contrôle des appels à partir du noeud origine. Pour bien illustrer l'idée de l'équivalence entre les tables de priorités sous les deux types de politiques, nous nous référerons à la figure 4.4.

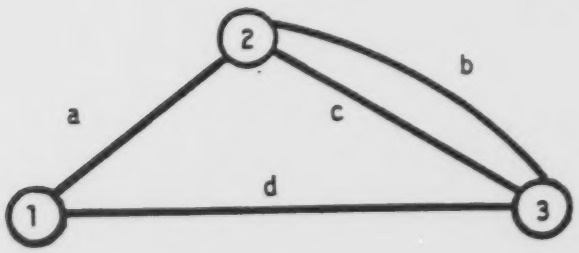


Figure 4.4 Réseau exemple.

Le tableau 4.1 contient les priorités de routage lorsque les décisions sont prises à chaque noeud. L'élément (i,j) de la table indique la priorité de sélection du $j^{ième}$ arc pour un appel rendu au $i^{ème}$ noeud.

TABLEAU 4.1

Noeud \ Arc	a	b	c	d
1	1	—	—	2
2	—	1	2	—

Le tableau 4.2 montre la table de priorité équivalente à 4.1 mais sous une politique de contrôle à partir du noeud origine de l'appel.

TABLEAU 4.2

Origine \ Chemin	(a,b)	(a,c)	d
1	1	2	3

Le tableau 4.2 a été fabriquée en utilisant le tableau 4.1 de façon lexicographique. L'ordre de priorité des chemins à emprunter à partir du noeud origine est établi en énumérant un arbre lexicographique à partir du premier tableau. Les décisions prises avec l'une ou l'autre table sont équivalentes à cause de l'aspect statique du réseau, c.-à-d. les priorités ne varieront pas en fonction du temps où le choix de routage est déterminé.

Dans les prochaines sections, nous allons montrer des modèles "avec ordonnancement" et de "croissance proportionnelle". Le modèle avec ordonnancement est un modèle avec débordement où l'ordre des chemins à utiliser pour chaque paire O/D a été pré-établie.

Avant de rentrer dans la description des modèles, il est important de mentionner leurs restrictions. D'abord, il ne s'agira pas de modèles dynamiques, c'est à dire, que la demande ne changera pas pendant la tranche de temps choisie pour étudier le routage. Il est à mentionner, cependant, qu'il serait possible d'utiliser les mêmes modèles, avec certaines modifications, pour l'étude du routage dynamique. Le deuxième point à considérer est encore relié à la notion de temps. On suppose

que les modèles sont appliqués pendant un temps assez court pour qu'une demande indépendante du temps soit représentative. Par exemple, nous pouvons prendre une période type du matin, celle de l'heure de pointe, du soir, etc.

Pour conclure cette introduction nous jugeons pertinent de ré-introduire un terme qui simplifiera la terminologie. Il s'agit du terme "marché" pour indiquer une paire de noeuds Origine-Destination. Nous mentionnons aussi que chaque marché r aura n_r choix de routes, à moins que d'autres spécifications soient données, et que chaque modèle acheminera les appels sur ces routes selon une politique différente.

4.3.3.1 Modèle MS3 (Modèle avec ordonnancement)

Nous voulons établir une ordre de débordement par marché où l'on utilise le premier choix de route tant que la route ne déborde pas, le débordement de la première route essayant de passer sur la deuxième, le débordement de la deuxième essayant sur la troisième et ainsi de suite jusqu'à ce que la $n^{ième}$ route soit remplie. Le débordement de cette dernière route passera sur une route fictive de capacité infinie.

Soit:

X_q=flot de trafic offert à la route q,

H_q=flot de trafic accepté sur la route q,

Q_r=ensemble des routes appartenant au marché r,

$P_{qr}(X_q)$= flot de débordement de la route $q \in Q_r$,

n_r=cardinalité de Q_r,

alors la figure 4.5 montre le comportement du modèle de façon schématique.

Nous mentionnons que l'expression de la fonction de débordement P_{qr} peut être déduite de l'expression $P(X_q^r)$ utilisée dans les modèles d'optimisation globale (eq. 4.3). Nous laissons les détails de dérivation de la fonction P_{qr} pour une section subséquente et passons immédiatement à la mise en équation du modèle.

Si $n_r = 3$ pour tout $r \in R$ alors

$$\begin{aligned} X_q &= H_q + P_{qr}(X_q) \qquad q = 1, 2, 3, 4 \\ X_1 &= d^r \\ X_2 &= P_{1r}(X_1) \\ X_3 &= P_{2r}(X_2) \\ X_4 &= P_{3r}(X_3). \end{aligned} \tag{4.12}$$

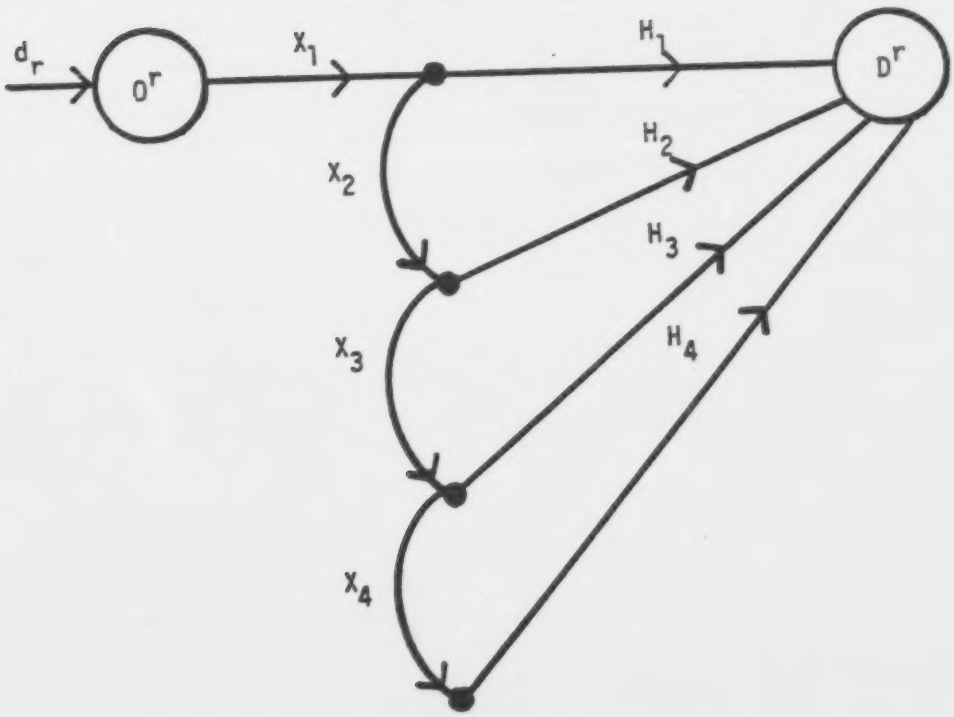


Figure 4.5 Politique de routage du modèle MS3

Le système d'équations non-linéaires (4.12) peut être ré-écrit de façon plus compacte comme suit:

$$
\begin{aligned}
H_1 + H_2 + H_3 + H_4 &= d^r \\
H_2 + H_3 + H_4 &= P_{1r}(d^r) = X_2 \\
H_3 + H_4 &= P_{2r}\left(P_{1r}(d^r)\right) = X_3 \\
H_4 &= P_{3r}\left(P_{2r}\left(P_{1r}(d^r)\right)\right) = X_4 \\
&\qquad \forall r \in R.
\end{aligned}
\tag{4.13}
$$

Les deux systèmes présentent huit équations et huit inconnus, (4.12) de façon explicite et (4.13) de façon implicite. La figure 4.6 montre la répartition du flot sur les trois routes étant donné les équations proposées.

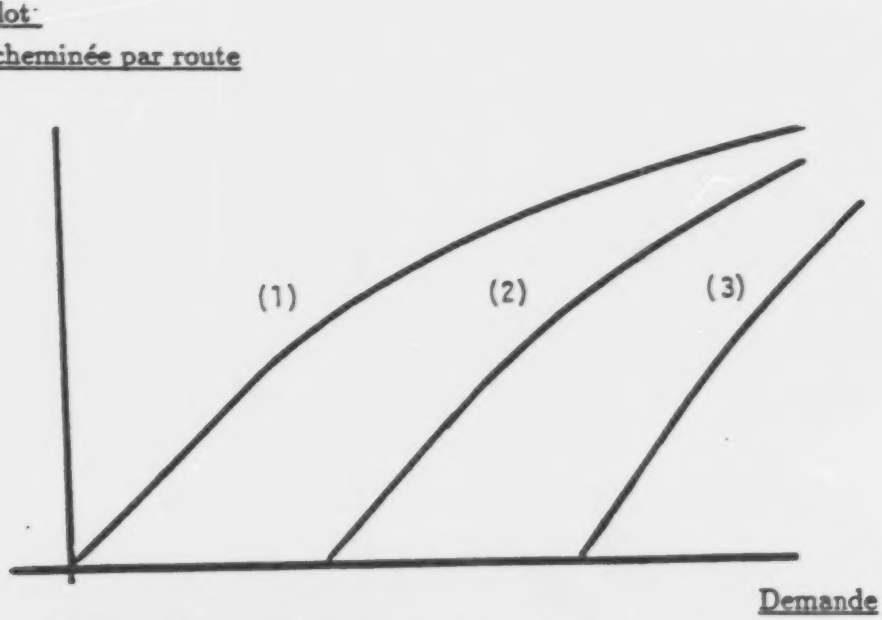


Figure 4.6 Répartition du flot sur les trois routes d'après le modèle MS3.

4.3.3.2 Description de la fonction de débordement

Lorsque l'on traite la fonction P_{qr} à l'intérieur du modèle MS3, trois problèmes sont identifiés.

D'abord on sait que P_{qr} correspond à la fonction du flot de débordement par route et nous connaissons seulement l'expression de $P(X)$, la fonction de débordement par arc. Il s'ensuit qu'il est nécessaire de déduire P_{qr} de $P(X)$ ce qui, nous verrons, n'est pas une tâche facile.

A part le fait que $P(X)$ et $P_{qr}(X)$ sont calculées par arc et par route respectivement, il existe une autre différence majeure entre les deux expressions. La première évalue le débordement total sur les arcs sans tenir compte du marché auquel ce dernier appartient. P_{qr}, par contre, correspond au débordement propre au marché r sur la route q. Maintenant, si l'on ne pose aucune restriction au partage des routes par plusieurs marchés, nous pouvons avoir une situation où plusieurs routes, du même marché ou non, ont des arcs en commun. Ils s'ensuit que si P_{qr} est calculé en fonction de $P_u(X)$, il y aura probablement au moins un arc de la route q où, soit le trafic offert, soit le débordement appartient à des marchés différents. Cette situation complique beaucoup le calcul de P_{qr}.

L'utilisation de la formule (4.3) du débordement total du réseau, $P(X)$, suppose que le trafic offert à l'arc u est poissonien. Le modèle MS3 établit que le trafic offert à une route équivaut au débordement de la route précédente. Ce fait considère, implicitement, que les débordements des routes sont poissoniens aussi, ce qui n'est pas le cas. Nous sommes conscients de l'erreur introduite en faisant cette hypothèse, nous invoquons cependant l'idée suivante: l'on peut considérer que les processus de débordement de différents types de flot sont des processus de renouvellement indépendants et identiquement distribués. Il s'ensuit que la superposition de tels processus tend vers un processus de Poisson. Évidemment, tel que l'on a déjà mentionné, l'indépendance des processus de débordement constitue, elle aussi, une hypothèse qui n'est vérifiée que dans des circonstances très particulières.

Afin de trouver les expressions pour P_{qr}, représentons, dans un contexte plus formel (4.3):

$$P(X) = \sum_{u \in M} P_u(X) = \sum_{u \in M} \sum_{p \in O} X_u^p B(C_u, \sum_{p \in O} X_u^p). \tag{4.3}$$

Nous avons déjà mentionné que (4.3) représente une borne supérieure de la valeur totale des pertes dans le réseau. De façon équivalente, nous pouvons exprimer la fonction de pertes sur une route $P_q(X)$ comme une somme des pertes sur tous les arcs de la route:

$$P_q(X) = \sum_{u \in q} P_u(X). \tag{4.14}$$

L'expression (4.14) renferme une hypothèse, appelée hypothèse A,qui s'annonce comme suit:

"*la probabilité qu'il y ait plus d'un arc bloqué sur un chemin est négligeable*".

Nous avons déjà mentionné que cette hypothèse n'est pas toujours vérifiée dans des réseaux réels, il s'ensuit qu'il est parfois nécessaire d'utiliser la formule suivante:

$$P_q(X) = X_q \left(1 - \prod_{u \in q} (1 - B(C_u, X_u)) \right) \tag{4.15}$$

où X_q dénote le flot sur la route q. L'équation (4.15) est valable seulement si les flots non acheminés sur les arcs sont indépendants les uns des autres. En effet, moins le flot sur les arcs est dépendant de celui de la route, (plusieurs routes partagent le même arc) plus petite sera la dépendance entre flots. D'un autre côté, si tous les arcs de la route en étude ne sont utilisés que par cette route, on aura la plus grande dépendance entre flots. La probabilité de refus sur la route q est donnée par: $\max_{u \in q} B(C_u, x_u)$.

Retournons au contexte à plusieurs marchés où il existe des arcs communs à différentes paires Origine-Destination. Si l'on remplace $P_q(X)$ par son équivalent $P_{qr}(X)$ dans la formule (4.15), il y aura un autre type de dépendance présente. Il s'agit de l'interaction entre le flot de différents marchés à l'intérieur des arcs en commun. Le problème des dépendances n'est pas facile à résoudre et il ne sera pas traité dans cet étude.

Pour utiliser (4.14) lorsqu'il y a plusieurs marchés, il faut faire une hypothèse en relation aux probabilités de débordement de plusieurs produits empruntant le même arc. Cette hypothèse, que nous allons appeler hypothèse B, est énoncée comme suit:

"*la probalilité de blocage sur un arc ne varie pas avec le type de produit qui essaie d'emprunter cet arc*".

L'hypothèse précédente obéit à un comportement "premier arrivé, premier servi" sur l'arc. Elle ne s'avère pas complètement vraie car il peut exister un blocage hétérogène de trafic qui est fait en fonction des caractéristiques différentes de appels qui ont déjà été bloquées ou de ceux qui ont écolué.

Si l'hypothèse B est vérifiée, alors la fonction de débordement par route et par marché peut être exprimée par:

$$P_{qr} = \sum_{u \in q} \frac{X_u^r}{X_u} P_u(X). \tag{4.16}$$

L'équation (4.16) représente donc une somme pondérée du refus sur chaque arc d'une route, son utilisation est alors assujettie aux mêmes restrictions que (4.13), par exemple l'hypothèse A doit être vérifiée.

4.3.3.3 Modèle de croissance proportionelle (MS4)

Dans ce modèle, nous supposons que chaque route à l'intérieur de chaque marché a un coefficient d'attraction A_q qui lui a été assigné. Notons que A_q peut être établi en fonction des tables de routage dont on a parlé dans l'introduction de la section. L'idée du modèle a été pour la première fois exposée par

Soumis (1978) en transport aérien. Ce modèle reprend le même principe que les modèles sans débordement avec partage de charge mais avec la notion de ré-essaie qui fait qu'il n'y a pas de véritable blocage.

La figure 4.7 montre la politique de routage à l'intérieur du marché r pour $n_r = 3$.

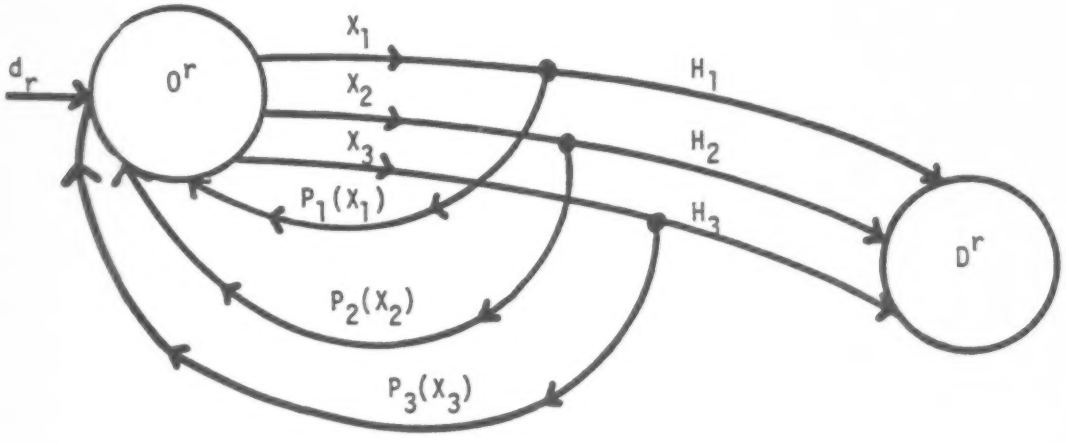


Figure 4.7 Politique de routage du modèle MS4

Selon la figure 4.7, la demande du marché d^r est répartie sur les trois routes du marché en proportion aux coefficients d'attraction des routes A_q. Une partie sera acceptée, la partie refusée ré-essaie et est toujours répartie en fonction des coefficients d'attraction.

D'après le comportement décrit, il existe une variable u_r pour chaque route $q \in Q_r$ telle que

$$X_q = u_r A_q \tag{4.17}$$

où u_r est le nombre total d'essaies du marché r. L'équation (4.17) équivaut à:

$$H_q + P_{qr}(X_q) = u_r A_q. \tag{4.18}$$

La fonction P_{qr} est calculée en prenant les hypothèses énoncées dans la section précédente. Maintenant, pour pouvoir écrire les équations du modèle en fonction d'un même type de variable, nous devons exprimer P_{qr} comme une fonction du flot accepté $\mathcal{H}$ et non du trafic offert $\mathcal{X}$. Nous utilisons les symboles $\mathcal{H}$ et $\mathcal{X}$ pour indiquer des vecteurs où les composantes sont les flots sur toutes les routes du réseau. Bien que ce ne soit pas un calcul facile à réaliser, il semble possible de trouver une fonction s telle que $s(\mathcal{H}) = \mathcal{X}$ et reformuler (4.18) en termes de $\mathcal{H}$.

Les équations du modèle MS4 sont alors, $\forall\, r \in R$:

$$\frac{H_q + P_{qr}\big(s(\mathcal{H})\big)}{A_q} = u_r, \tag{4.19}$$

$$\sum_{q \in Q_r} H_q = d^r. \tag{4.20}$$

La figure 4.8 montre la répartition du flot sur un marché avec $n_r = 3$.

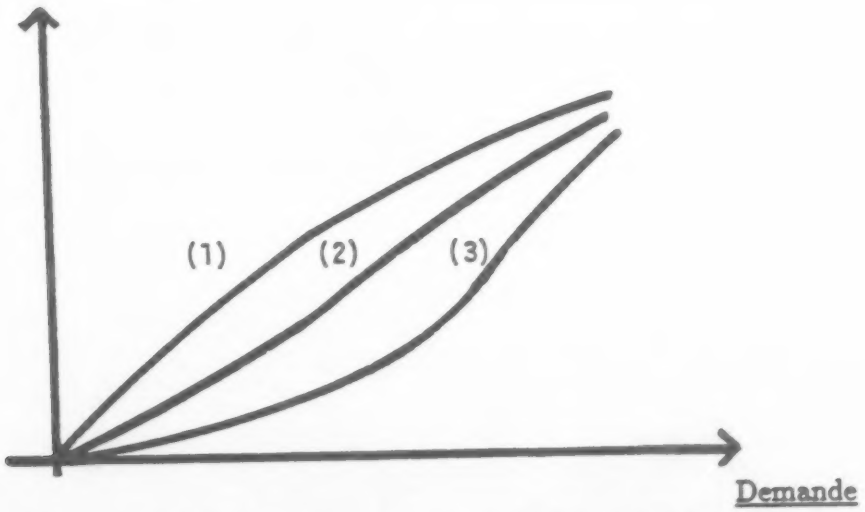


Figure 4.8 Répartition du flot selon le modèle MS4

4.3.4 Équivalence entre les modèles proposés et des modèles d'optimisation globale

Il a déjà été mentionné que, dans certains cas, les équations qui décrivent le comportement décentralisé d'un marché correspondent aux conditions de Kuhn-Tucker d'un algorithme d'optimisation globale. Il s'ensuit qu'il est parfois souhaitable de chercher la correspondance entre les deux formulations et de résoudre le problème d'optimisation équivalent plutôt que le système d'équations original.

Le but de cette section est d'essayer de trouver deux problèmes d'optimisation dont les conditions de Khun-Tucker correspondent aux équations des modèles MS3 et MS4 respectivement.

Commençons, pour des raisons de simplicité, par le modèle de croissance proportionelle MS4. Après inspection des équations (4.19) et (4.20) du modèle, nous proposons intuitivement le problème (OP) montré ci-dessous comme un bon candidat à être l'équivalent, en optimisation, de MS4.

Soit (OP) :

$$\min f(L) \tag{4.21}$$

s. à

$$\sum_{q \in Q_r} L_q = d^r \qquad \forall r \in R. \tag{4.22}$$

Les conditions de Kuhn-Tucker de ce problème sont:

$$\frac{\partial f}{\partial L_q} = u_r \qquad \forall q \in Q_r \quad \forall r \in R \tag{4.23}$$

$$\sum_{q \in Q_r} L_q = d^r. \tag{4.24}$$

Pour que le système d'équations (4.23) et (4.24) soit équivalent au système définit par les équations (4.19) et (4.20), les variables L et H doivent être égales et il faut vérifier l'équation suivante pour f étant une fonction continue:

$$\frac{H_q + P_{qr}(s(\mathcal{H}))}{A_q} = \frac{\partial f(L)}{\partial L_q}. \tag{4.25}$$

Une condition nécessaire de continuité de f est donnée par:

$$\frac{\partial}{\partial L_p}\left(\frac{\partial f}{\partial L_q}\right) = \frac{\partial}{\partial L_q}\left(\frac{\partial f}{\partial L_p}\right) \qquad \forall p, \forall q, \forall r \tag{4.26}$$

ou, en substituant (4.25) dans (4.26):

$$\frac{\partial}{\partial H_p}\left(\frac{H_q + P_{qr}(s(\mathcal{H}))}{A_q}\right) = \frac{\partial}{\partial H_q}\left(\frac{H_{p+}P_{pr}(s(\mathcal{H}))}{A_p}\right). \qquad \forall p, \forall q, \forall r \tag{4.27}$$

L'équation (4.27) n'est pas satisfaite dans tous les cas, mais elle est satisfaite s'il n'y a pas d'arcs en commun entre les routes appartenant à des marchés différents.

Cependant, si l'on travaille à l'intérieur de chaque marché, la condition (4.27) est vérifiée. Il serait donc possible d'étudier le comportement de chaque marché comme un problème d'optimisation tel que (OPM_r).

Soit (OPM_r) :

$$\min \sum_{q \in Q_r} \int \left[\frac{H_q + P_q(s(\mathcal{H}))}{A_q}\right] dH_q \tag{4.28}$$

s. à

$$\sum_{q \in Q_r} H_q = d^r. \tag{4.29}$$

Pour ce qui est du modèle MS3, ré-écrivons les équations (4.12) sous la forme suivante:

$$X_4 = P_{3r}(X_3) = P_{3r}\big(P_{2r}(X_2)\big) = P_{3r}\Big(P_{2r}\big(P_{1r}(X_1)\big)\Big), \tag{4.30}$$

$$H_1 + H_2 + H_3 + H_4 = d^r \qquad \forall r \in R \tag{4.31}$$

alors, pour que (4.34) soit équivalent à (4.28), les variables L et H doivent être égales et $\forall r \in R$ le système d'équations doit être vérifié:

$$X_4 = \frac{\partial f(H)}{\partial H_4}, \tag{4.32}$$

$$P_{3r}(X_3) = \frac{\partial f(H)}{\partial H_3}, \tag{4.33}$$

$$P_{3r}\big(P_{2r}(X_2)\big) = \frac{\partial f(H)}{\partial H_2}, \tag{4.34}$$

$$P_{3r}\Big(P_{2r}\big(P_{1r}(X_1)\big)\Big) = \frac{\partial f(H)}{\partial H_1}. \tag{4.35}$$

Si les équations (4.32) à (4.35) s'avèrent vraies, il faut toujours se demander si la fonction f est continue. Nous avons vu que pour le modèle de croissance proportionelle il n'est pas possible d'assurer l'existence d'une fonction continue $f(H)$. Or, si les conditions de continuité ne peuvent pas être vérifiées dans un modèle sans ordonnancement, ceci étant dû à l'interaction entre marchés, il est encore moins probables de les vérifier dans un modèle avec ordonnancement du débordement, c.-à-d. où l'interaction entre marchés est beaucoup plus forte. Nous rejetons donc l'idée de trouver une fonction continue $f(H)$ pour le modèle MS4.

La question que l'on se pose maintenant est si, à l'intérieur de chaque marché, il serait possible de trouver un problème d'optimisation dont les conditions de Kuhn-Tucker sont les équations du modèle pour le marché en question. Ceci est équivalent à fixer la valeur de r dans les équations (4.30) à (4.35).

Puisque les équations (4.32) à (4.35) constituent un système avec des variables de nature différentes, il est nécessaire d'exprimer les P comme fonctions des variables H. Ceci peut être fait uniquement si la fonction inverse $s(\mathcal{H})$ peut être trouvée. Naturellement, dû à la nature d'ordonnancement du modèle, la fonction $s(\mathcal{H})$ est trouvable uniquement lorsqu'il n'y a pas de routes, même à l'intérieur d'un même marché, qui partagent des arcs. Il s'agit, évidemment, d'une condition tout à fait imaginaire, car elle impliquerait qu'il existe plusieurs routes dédiées à chaque marché sans qu'il n'y ait aucun partage de liens dans le réseau. Cependant, sous cette condition, le modèle d'optimisation équivalent aux équations (4.30) et (4.31) pour chaque marché r serait

(OPM_r) :

$$\begin{aligned} \min \frac{H_4^2}{2} &+ \int P_{3r}(s(H_3))\, dH_3 + \\ &+ \int P_{3r}\Big(P_{2r}(s(H_2))\Big)\, dH_2 + \\ &+ \int P_{3r}\Big(P_{2r}\Big(P_{1r}(s(H_1))\Big)\Big)\, dH_1 \end{aligned} \tag{4.36}$$

s. à

$$H_1 + H_2 + H_3 + H_4 = d^r. \tag{4.37}$$

Dans la section 4.2, nous avions remarqué que les modèles avec partage de charge sont plus proches de l'optimisation que les modèles avec débordement. Nous avons aussi mentionné dans les sections respectives que MS3 est un modèle avec débordement et que MS4 ressemble à un modèle de partage de charge. Il n'est donc pas surprenant que, tel que l'on a montré dans cette section, les conditions pour que MS3 puisse être résolu par un programme d'optimisation soient beaucoup plus restrictives que les conditions pour résoudre MS4 avec un algorithme d'optimisation globale.

4.4 Algorithme de routage

Le modèle de routage que nous avons décidé de mettre en place a été MS2, c.-à-d., un modèle stochastique global où chaque produit représente les appels originant d'un noeud.

Nous allons donner les détails du réseau test utilisé dans le chapitre des résultats. Nous avançons, cependant, qu'il s'agit d'un réseau de 40 noeuds et 62 arcs non orientés. Pour une telle taille de réseau nous avions le choix entre deux méthodes de résolution: la simplexe convexe et l'algorithme de Frank-Wolfe.

Au niveau du routage les deux algorithmes d'optimisation devraient donner des temps de calcul comparables pour la taille du problème considéré. C'est au niveau de la réoptimisation (c.-à-d. le routage pour les états de panne) qu'il y aurait une différence si l'on utilise l'un ou l'autre algorithme, le simplexe convexe étant beaucoup plus efficace au niveau de la réoptimisation que l'algorithme de Frank-Wolfe.

Nous avons, malgré la supériorité algorithmique de la Simplexe Convexe pour la réoptimisation, choisit la méthode de Frank-Wolfe pour des raisons de simplicité de programmation. En effet, le simplexe convexe n'étant pas disponible, un code de Frank-Wolfe peut être implanté en quelques semaines. De plus, l'idée de ce travail était de développer un prototype pour montrer l'interaction entre le routage et l'évaluation de la performance du réseau, les raffinements algorithmiques étant laissé pour des projets futurs. Finalement, nous verrons dans le prochain chapitre que, si la disponibilité des arcs est très élevée, comme dans le plupart des vrais réseaux de télécommunications, les résultats des états de panne n'auront pas beaucoup d'influence sur la mesure de performance. Il s'ensuit qu'il n'est pas nécessaire d'être très exigeant au niveau du reroutage ce qui justifierait encore plus l'utilisation de l'algorithme de Frank-Wolfe.

Nous n'allons pas décrire les détails de l'algorithme de Frank-Wolfe, on se limitera à mentionner que l'implantation correspond à celle d'une méthode de Frank-Wolfe "classique" semblable à celles étudiées dans le problèmes d'affectation du trafic d'équilibre (voir Guélat (1983)). Nous allons, cependant, allouer toute

une section à la convergence de l'algorithme pour le type de problème traité; le résultat principal de la section est la preuve de convexité de la fonction objectif ce qui constitue un résultat assez important en télécommunications.

4.4.1 Convergence de l'algorithme

La convergence de l'algorithme vers un optimum global est garantie si la fonction objectif est convexe. Il s'ensuit que l'étude de la convergence de la méthode sera limitée à la preuve de convexité de la fonction objectif.

La fonction objectif utilisée dans l'implantation du modèle de routage MS2 est donnée par:

$$\sum_{u \in M} \Big(K\mathcal{L}(X_u) + \mathcal{P}(X_u) + M\mathcal{P}^2(X_u) \Big). \tag{4.38}$$

Dans (4.38), X_u représente la demande totale sur l'arc u, $\mathcal{L}(X_u)$ est un terme linéaire de plus courts chemins, $\mathcal{P}(X_u)$ est la fonction de pénalité et K et M sont des poids donnés aux termes linéaires et quadratiques, respectivement. La preuve de convexité de la fonction objectif est donc limitée à la preuve de convexité de la fonction de pénalité. Nous allons ré-écrire, de façon explicite le terme de pénalité (eq. 4.39)

$$P(X) = \sum_{u \in M} \mathcal{P}(X_u) = \sum_{u \in M} \sum_{p \in O} k^p X_u^p \; B(C_u, \sum_{p \in O} X_u^p). \tag{4.39}$$

Dans (4.39), la fonction B qui représente la probabilité de refus sur un arc peut être exprimé par:

$$B(C_u, \sum_{p \in O} X_u^p) = \frac{\left(\sum_{p \in O} X_u^p \right)^{C_u} / C_u!}{\sum_{k=0}^{C_u} \left(\sum_{p \in O} X_u^p \right)^{k} / k!}. \tag{4.40}$$

Nous considérons ici seulement le cas où $k^p = 1$, alors on peut faire la substitution suivante:

$$X_u = \sum_{p \in O} X_u^p. \tag{4.41}$$

Et, finalement la fonction de l'espérance d'unités d'appels refusés dans le réseau peut être exprimée par:

$$P(X) = \sum_{u \in M} \frac{X_u^{C_u+1}/C_u!}{\sum_{k=0}^{C_u} X_u^k/k!}. \tag{4.42}$$

Comme $P(X)$ est séparable par arcs, la convexité de P par rapport à la demande totale sur les arcs X_u sera démontrée si $\mathcal{P}(X_u)$, la fonction de refus sur chaque arc, est convexe. La preuve du théorème de convexité de $\mathcal{P}(X_u)$ sera réalisée de la façon suivante: d'abord nous allons ré-arranger les termes de $\mathcal{P}(X_u)$, ensuite on montrera la condition nécessaire et suffisante de convexité et troisièmement l'on démontrera plusieurs lemmes nécessaires à la preuve du théorème.[1]

Reprenons, tout d'abord, l'expression de $\mathcal{P}(X_u)$ où, par simplicité, on a omis les sous-indices relatifs aux arcs:

$$\mathcal{P}(X) = \frac{X^{C+1}/C!}{\sum_{k=0}^{C} X^k/k!} \qquad C \text{ entier} \tag{4.43}$$

$\mathcal{P}(X)$ peut s'écrire comme l'inverse d'une fonction $\mathcal{F}(X)$ déduite de (4.43).

$$\mathcal{F}(X) = \frac{1}{\mathcal{P}(X)} = \sum_{t=0}^{C} \frac{C!}{(C-t)!} \frac{1}{X^{t+1}}. \tag{4.44}$$

À partir de (4.44) l'on déduit les dérivées première et deuxième de la fonction $\mathcal{F}$:

$$\mathcal{F}'(X) = \sum_{t=0}^{C} \frac{C!}{(C-t)!} \frac{(-(t+1))}{X^{t+2}} \tag{4.45}$$

[1] Une partie de la preuve que nous présentons a été réalisée par Pierre Trudeau du Centre de Recherche sur les Transports.

et

$$\mathcal{F}''(X) = \sum_{t=0}^{C} \frac{C!}{(C-t)!} \frac{(t+1)(t+2)}{X^{t+3}}. \tag{4.46}$$

On voit facilement que $\mathcal{F}'(X) \leq 0$ et $\mathcal{F}''(X) \geq 0$.

En exprimant $\mathcal{P}''(X)$ en fonction de $\mathcal{F}$ et ses dérivées on obtient que $\mathcal{P}''(X) \geq 0$ si et seulement si

$$2\Big(\mathcal{F}'(X)\Big)^2 \geq \mathcal{F}''(X)\,\mathcal{F}(X). \tag{4.47}$$

En effectuant les produits de séries présents de chaque côté de l'inégalité (4.47) on obtient, pour le côté gauche de l'expression (4.47)

$$2\left[\sum_{m=0}^{C}\sum_{k=0}^{m} \frac{C!C!}{(C-k)!(C-m+k)!} \frac{(k+1)(m-k+1)}{X^{m+4}} + \sum_{m=C+1}^{2C}\sum_{k=m-C}^{C} \frac{C!C!}{(C-k)!(C-m+k)!} \frac{(k+1)(m-k+1)}{X^{m+4}}\right] \tag{4.48}$$

et pour celui de droite

$$\sum_{m=0}^{C}\sum_{k=0}^{m} \frac{C!C!}{(C-k)!(C-m+k)!} \frac{(k+1)(k+2)}{X^{m+4}} + \sum_{m=C+1}^{2C}\sum_{k=m-C}^{C} \frac{C!C!}{(C-k)!(C-m+k)!} \frac{(k+1)(k+2)}{X^{m+4}}. \tag{4.49}$$

Nous allons au cours de la preuve, introduire des notations qui nous permetteront de simplifier de longues expressions. Soit, par example, $p(k)$ l'expression suivante:

$$p(k) = \frac{C!C!}{(C-k)!(C-m+k)!} \qquad \forall\, k,\, \forall\, m. \tag{4.50}$$

Nous avons tous les éléments pour énoncer le premier lemme.

Lemme 1. *La fonction $\mathcal{P}$ sera convexe si les inégalités suivantes sont satisfaites*

$$\sum_{k=0}^{m} p(k)\,(k+1)\,(2m-3k) \geq 0 \qquad m = 1, \ldots, C \tag{4.51}$$

et

$$\sum_{k=m-C}^{C} p(k)\,(k+1)\,(2m-3k) \geq 0 \qquad m = C+1, \ldots, 2C. \tag{4.52}$$

Preuve.
Substituons les expressions (4.48) et (4.49) dans l'inégalité (4.47) et, en additionnant les coefficients des termes qui présentent les mêmes puissances en X, on obtient les inégalités (4.51) et (4.52). ∎

Nous passons maintenant à l'étude d'une symétrie qui existe entre certains termes de la fonction $p(k)$. La symétrie est annoncée par les lemmes 2 et 3.

Lemme 2. $p(t) = p(m-t) \qquad \forall\, t = 0, 1, \ldots m$
$\forall\, m = 1, \ldots C.$

Preuve.
En substituant k par t dans l'expression de $p(k)$ on obtient

$$p(t) = \frac{C!C!}{(C-t)!(C-m+t)!}. \tag{4.53}$$

Si, maintenant, on substitue k par $m-t$, nous avons:

$$p(m-t) = \frac{C!C!}{(C-m+t)!(C-t)!} \tag{4.54}$$

Les expressions (4.53) et (4.54) sont équivalentes. ∎

Avant d'énoncer le lemme 3, introduisons la variable d:

$$d = m - C. \tag{4.56}$$

Alors, le Lemme 3 s'énonce comme suit

Lemme 3. $p(d+t) = p(C-t) \qquad \forall\, t = 0, \ldots, C-d$
$\forall\, d = 1, \ldots C.$

Preuve.
On substitue encore une fois k par $(d+t)$ et par $(C-t)$ dans (4.50) et l'on obtient les expressions équivalentes (4.57) et (4.58).

$$p(d+t) = \frac{C!C!}{(C-d-t)!t!}, \tag{4.57}$$

$$p(C-t) = \frac{C!C!}{t!(C-d-t)!}. \tag{4.58}$$

∎

Avant d'énoncer le prochain lemme nous introduisons la notation suivante:

$$p_1(t) = p(t) \qquad \forall\, t = 0, \ldots m, \tag{4.59}$$
$$\forall\, m = 1, \ldots C,$$

$$p_2(t) = p(d-t) \qquad \forall\, t = 0, \ldots C-d, \qquad (4.60)$$
$$\forall\, d = 1, \ldots C,$$

$$T(t) = (t+1)(2m-3t) \qquad \forall\, t = 0, \ldots m, \qquad (4.61)$$
$$\forall\, m = 1, \ldots 2C,$$

$$D_1(t) = T(t) + T(m-t) \qquad \forall\, t = 0, \ldots m, \qquad (4.62)$$
$$\forall\, m = 1, \ldots C,$$

$$D_2(t) = T(d+t) + T(C-t) \qquad \forall\, t = 0, \ldots m, \qquad (4.63)$$
$$\forall\, d = 1, \ldots C.$$

Alors, les lemmes 2 et 3 nous permettent de ré-écrire les termes de gauche des inégalités (4.51) et (4.52) sous la forme des expressions qui suivent:

$$\left[\left(\sum_{t=0}^{\tau_1} p_1(t)\, D_1(t) \right) + \gamma_1 p_1 \left(\frac{m}{2} \right) T \left(\frac{m}{2} \right) \right], \qquad (4.64)$$

$$\left[\left(\sum_{t=0}^{\tau_2} p_2(t)\, D_2(t) \right) + \gamma_2 p_2 \left(\frac{C+d}{2} \right) T \left(\frac{C+d}{2} \right) \right]. \qquad (4.65)$$

Il est à noter que dans (4.65) la variable m a été remplacée par son équivalent $(C+d)$.

Dans les expressions précédentes nous avons:

$$\tau_1 = \frac{m-1}{2} \text{ et } \gamma_1 = 0 \text{ si } m \text{ impair}, \qquad (4.66)$$

$$\tau_1 = \frac{m-2}{2} \text{ et } \gamma_1 = 1 \text{ si } m \text{ pair}, \qquad (4.67)$$

$$\tau_2 = \frac{(C-d)-1}{2} \quad \text{et } \gamma_2 = 0 \text{ si } (C-d) \text{ impair}, \qquad (4.68)$$

$$\tau_2 = \frac{(C-d)-2}{2} \quad \text{et } \gamma_2 = 1 \text{ si } (C-d) \text{ pair}. \qquad (4.69)$$

Nous allons, maintenant, énoncer un résultat à propos des fonctions $p_1(t)$ et $p_2(t)$ sous la forme de lemme.

Lemme 4. *Les fonctions $p_1(t)$ et $p_2(t)$ sont croissantes en t.*

Preuve.
Il s'agit de comparer le $t^{ième}$ avec le $(t+1)^{ième}$ terme pour chaque fonction. Nous avons pour $p_1(t)$:

$$\frac{C!C!}{(C-t)!(C-m+t)!} \leq \frac{C!C!}{(C-t-1)!(C-m+t+1)!} \tag{4.70}$$

qui équivaut à

$$C-m+t+1 \leq C-t \tag{4.71}$$

ou, en re-grouppant les termes

$$2t-m+1 \leq 0. \tag{4.72}$$

L'inéquation précédente doit être vérifiée lorsque $t = 0,\ldots,\tau_1$, donc il suffit de la vérifier pour le t maximal, c.-à-d. $\frac{(m-1)}{2}$. En substituant cette valeur dans (4.72) on obtient:

$$m-1-m+1=0 \leq 0 \tag{4.73}$$

ce qui démontre la première partie du lemme.

Pour ce qui est de la fonction $p_2(t)$, on doit satisfaire

$$\frac{C!C!}{(C-d-t)!t!} \leq \frac{C!C!}{(C-d-t-1)!(t+1)!}. \tag{4.74}$$

Ce qui équivaut à

$$2t + 1 + d \leq C. \tag{4.75}$$

Vérifions pour le t maximal, $\frac{(C-d)-1}{2}$, et nous avons

$$(C - d) - 1 + 1 + d \leq C. \tag{4.76}$$

Ce qui conclut la preuve du lemme. ∎

Passons, maintenant, au lemme 5.

Lemme 5. *Les expressions suivantes sont toujours non négatives:*

$$\left(\sum_{t=0}^{r_1} D_1(t)\right) + \gamma_1 T\left(\frac{m}{2}\right), \tag{4.77}$$

$$\left(\sum_{t=0}^{r_2} D_2(t)\right) + \gamma_2 T\left(\frac{C+d}{2}\right). \tag{4.78}$$

Preuve.
Nous remarquons, d'abord, que (4.77) et (4.78) correspondent, respectivement aux membres de gauche des inégalités (4.51) et (4.52) si l'on pose $p(k) = 1$. De façon explicite, nous savons que (4.77) équivaut à

$$\sum_{k=0}^{m} (k+1)(2m - 3k) \tag{4.79}$$

et (4.78) à

$$\sum_{k=d}^{C} (k+1)(2m - 3k). \tag{4.80}$$

Travaillons, d'abord avec (4.79); en multipliant et séparant les termes l'on obtient:

$$\sum_{k=0}^{m}(2m-3)k-3\sum_{k=0}^{m}k^2+2\sum_{k=0}^{m}m \tag{4.81}$$

à l'aide des relations bien connues (4.82) et (4.83)

$$\sum_{j=1}^{n}j=\frac{n(n+1)}{2}, \tag{4.82}$$

$$\sum_{j=1}^{n}j^2=\frac{n(n+1)(2n+1)}{6}, \tag{4.83}$$

l'on obtient, après certaines manipulations,

$$\frac{(m+1)}{6}m\big[6m-9-6m-3+12\big]=0 \tag{4.84}$$

d'où (4.77) est non négative.

Pour ce qui est de (4.78), reprenons l'équation (4.80) avec le changement de variables $k=k'+d$,

$$\sum_{k'=0}^{C-d}(d+k'+1)\Big(2(C+d)-3(k'+d)\Big). \tag{4.85}$$

En multipliant et séparant les termes de (4.85) nous obtenons

$$\sum_{k'=0}^{C-d}(2C-d-3k')(k'+1)+d\sum_{k'=0}^{C-d}(2C-d-3k') \tag{4.86}$$

et après

$$\sum_{k'=0}^{C-d}(2C-d-3k')(k'+1)+d\sum_{k'=0}^{C-d}(2C-2d+d-3k') \tag{4.87}$$

et

$$\sum_{k'=0}^{C-d}(2(C-d)-3k')(k'+1)+d\sum_{k'=0}^{C-d}(k'+1)+d\sum_{k'=0}^{C-d}(2C-d-3k'). \tag{4.88}$$

Nous avons démontré que l'expression (4.79) est égale à zéro pour tout m. Il s'ensuit que la première série de (4.88) s'annulle. En manipulant les termes restants nous obtenons:

$$d\sum_{k'=0}^{C-d}(2C+1)-d^2\sum_{k'=0}^{C-d}1-2d\sum_{k'=0}^{C-d}k' \tag{4.89}$$

et, en faisant appel encore une fois à l'identité (4.82) nous avons,

$$\frac{(C-d+1)}{2}\Big[2dC+2d\Big] \tag{4.90}$$

ce qui équivaut à:

$$d(C-d+1)(C+1), \tag{4.91}$$

(4.91) est toujours positive car la valeur maximale de d est C. ∎

Passons, maintenant au lemme suivant:

Lemme 6. *Les fonctions* $D_i(t)$ $(i = 1, 2)$ *données par les expressions (4.62) et (4.63) sont telles que*

$$D_i(t) < 0 \text{ si } t \in (-\infty, \beta_i^*) \tag{4.92}$$

et

$$D_i(t) \geq 0 \text{ si } t \in \left[\beta_i^*, \frac{\eta_i}{2}\right]. \tag{4.93}$$

Dans l'énoncé du lemme 6, η_i est défini comme m si $i = 1$ et comme $(C-d)$ si $i = 2$. β_i^* est une racine de $D_i(t)$ qui appartient à l'intervalle $(-\infty, \frac{\eta_i}{2}]$.

Preuve.
Prenons d'abord l'expression correspondante à $D_1(t)$.

$$D_1(t) = (t+1)(2m - 3t) + (m - t + 1)\big(2m - 3(m - t)\big). \tag{4.94}$$

En multipliant et regroupant des termes, nous trouvons un polynôme de deuxième degré en t.

$$D_1(t) = -6t^2 + 6mt + (m - m^2), \tag{4.95}$$

les racines de (4.95) sont données par

$$\frac{m}{2} \pm \frac{\sqrt{12m(m+2)}}{12}. \tag{4.96}$$

Il est facile de vérifier que $D_1(t)$ est une fonction concave croissante dans l'intervalle $(-\infty, \frac{m}{2})$. Il s'ensuit que dans l'intervalle $(-\infty, \beta_1^*)$ la fonction $D_1(t)$ est négative et que dans $[\beta_1^*, \frac{m}{2}]$ elle est positive, β_1^* étant donné par

$$\beta_1^* = \frac{m}{2} - \frac{\sqrt{12m(m+2)}}{12}. \tag{4.97}$$

Démontrons, maintenant le lemme pour la fonction $D_2(t)$.

$$\begin{aligned} D_2(t) =&(d+t+1)(2C+2d-3d-3t)+ \\ &(C-t+1)(2C+2d-3C+3t), \end{aligned} \tag{4.98}$$

ce qui donne le polynome suivant:

$$D_2 = -6t^2 + 6t(C-d) + (C+d-C^2-d^2+4Cd). \tag{4.99}$$

$D_2(t)$ a seulement une racine β_2^* dans l'intervalle $(-\infty, \frac{C-d}{2}]$ qui correspond à

$$\beta_2^* = \frac{(C-d)}{2} - \frac{\sqrt{12[(C+d+1)^2-1]}}{12} \tag{4.100}$$

et puisque $D_2(t)$ est aussi concave croissante dans l'ntervalle $(-\infty, \frac{C-d}{2}]$, le lemme est démontré. ∎

Théorème de Convexité. *La fonction $\mathcal{P}$ étant définie par l'équation (4.43) est convexe par rapport à X, la demande totale sur les arcs.*

Preuve du théorème.

Le lemme 1 démontre que les inégalités (4.51) et (4.52) garantissent la convexité de la fonction. Nous avons montré que les membres de gauche des inégalités (4.51) et (4.52) sont équivalents aux expressions (4.64) et (4.65) respectivement. Ces deux équations ainsi que la condition suffisante de convexité sont ré-exprimées comme:

$$\left(\sum_{t=0}^{r_i} p_i(t)\, D_i(t)\right) + \gamma_i p_i \left(\frac{m_i}{2}\right) T\left(\frac{m_i}{2}\right) \geq 0 \qquad i = 1, 2. \tag{4.101}$$

Dans (4.101) $m_i = m$ si $i = 1$ et $m_i = C + d$ si $i = 2$. Il s'ensuit que si (4.101) est vérifiée, l'on aura prouvé le théorème de convexité.

D'après le lemme 5 nous avons:

$$\sum_{t=0}^{\tau_i} D_i(t) + \gamma_i T\left(\frac{m_i}{2}\right) \geq 0 \quad i = 1, 2. \tag{4.102}$$

De plus, le terme $T\left(\frac{m_i}{2}\right)$ est toujours positif (il est facile de le vérifier en remplaçant $\frac{m_i}{2}$ dans (4.61)). D'un autre côté dans le lemme 6 on a déterminé que la fonction $D_i(t)$ est non négative seulement dans l'intervalle $(\beta_i^*, \frac{\eta_i}{2}]$. Maintenant, nous savons que $\beta_i^* \in (-\infty, \frac{\eta_i}{2}]$ mais on ne peut pas garantir que $\beta_i^* \in (-\infty, \tau_i]$, τ_i étant définit par les équations (4.66) à (4.69). Nous avons deux cas à étudier:

a) $\beta_i^* \in (-\infty, \tau_i]$

b) $\beta_i^* \in (\tau_i, \frac{\eta_i}{2}]$

Le cas **(a)** peut être sous-divisé en deux sous-cas:

a.1) $\beta_i^* \in (-\infty, 0)$

Dans ce cas-ci, $D_i(t)$ est positive sur tout l'intervalle $[0, \tau_i]$ et, puisque $p_i(t)$ et $T(\frac{m_i}{2})$ sont aussi positives, (4.102) est vérifiée.

a.2) $\beta_i^* \in [0, \tau_i]$

Définissons $\mathcal{S}^-$ comme la somme des coefficients négatifs de $D_i(t)$ et $\mathcal{S}^+$ comme la somme des positifs, alors

$$\mathcal{S}^- = \sum_{t=0}^{\lceil \beta_i^* \rceil - 1} D_i(t) \tag{4.103}$$

et

$$\mathcal{S}^+ = \sum_{t=\lceil \beta_i^* \rceil}^{\tau_i} D_i(t) + \gamma_i T\left(\frac{m_i}{2}\right), \tag{4.104}$$

alors, d'après (4.102) nous pouvons conclure que:

$$|\mathcal{S}^-| \leq \mathcal{S}^+. \tag{4.105}$$

Maintenant, pour que (4.105) soit satisfaite il faut que:

$$\left| \sum_{t=0}^{\lceil \beta_i^* \rceil - 1} p_i(t) D_i(t) \right| \leq \left(\sum_{t=\lceil \beta_i^* \rceil}^{\tau_i} p_i(t) D_i(t) \right) + \gamma_i p_i\left(\frac{m_i}{2}\right) T\left(\frac{m_i}{2}\right) \tag{4.106}$$

et, d'après le lemme 4 qui montre que les fonctions $p_i(t)$ sont croissantes, nous avons:

$$\left| \sum_{t=0}^{\lceil \beta_i^* \rceil - 1} p_i(t) D_i(t) \right| \leq p_i(\beta_i^*) \left| \sum_{t=0}^{\lceil \beta_i^* \rceil - 1} D_i(t) \right| = p_i(\beta_i^*) \left| \mathcal{S}^- \right| \tag{4.107}$$

et

$$p_i(\beta_i^*)|\mathcal{S}^-| \leq p_i(\beta_i^*)\mathcal{S}^+ \leq \left(\sum_{t=\lceil \beta_i^* \rceil}^{t_i} p_i(t) D_i(t) \right) + \\ + \gamma_i p_i\left(\frac{m_i}{2}\right) T\left(\frac{m_i}{2}\right) \tag{4.108}$$

ce qui démontre (4.106).

b) $\beta_i^* \in \left(\tau_i, \frac{\eta_i}{2}\right]$

Pour ce cas, $D_i(t)$ est négatif sur tout l'intervalle $[0, \tau_i]$, donc

$$\mathcal{S}^- = \sum_{t=0}^{\tau_i} D_i(t) \tag{4.109}$$

et, à partir de (4.102) nous trouvons que

$$\mathcal{S}^+ = \gamma_i T\left(\frac{m_i}{2}\right) \geq |\mathcal{S}^-|, \tag{4.110}$$

d'où

$$\left|\sum_{t=0}^{\tau_i} p_i(t) D_i(t)\right| \leq p_i(\tau_i)\,|\mathcal{S}^-| \tag{4.111}$$

et, puisque $\tau_i < \frac{m_i}{2}$ nous vérifions

$$p_i(\tau_i)|\mathcal{S}^-| \leq \gamma_i p\left(\frac{m_i}{2}\right) \mathcal{S}^+ = \gamma_i p\left(\frac{m_i}{2}\right) T\left(\frac{m_i}{2}\right), \tag{4.112}$$

d'où (4.101) est satisfaite pour le dernier cas et la preuve de la proposition 1 est complétée. ∎

Corollaire. *La fonction $P(X)$ est convexe par rapport à la variable X_u^p.*

Preuve du corollaire

D'après (4.39) $P(X)$ est convexe par rapport à X_u car il s'agit d'une somme de fonctions convexes. D'autre part, X_u est définit comme la somme des X_u^p, alors $P(X)$ est aussi convexe par rapport à X_u^p par la propriété de composition de fonctions convexes.

5. RÉSULTATS

Dans ce chapitre, nous allons montrer les résultats de l'application du modèle MS2 à un réseau test. Nous allons, tout d'abord, introduire le réseau et ses caractéristiques. Deuxièmement, nous parlerons du problème de l'évaluation de la fonction objectif. Dans un troisième temps, nous présenterons les résultats au niveau du routage et, finalement, les résultats reliés à la fiabilité. Chacune de ces étapes constitue une section du chapitre.

5.1 Présentation du réseau

Nous aurions voulu tester notre modèle sur un vrai réseau avec de vraies données. Malheureusement, pour des raisons de confidentialité des données des réseaux de télécommunications, nous n'avons pu utiliser un réseau existant. Malgré cela, on nous a fourni certains renseignements relatifs à un réseau existant. Ces renseignements nous ont permis de construire un prototype de réseau assez rapproché de la réalité. Les caractéristiques données par la compagnie sont énoncées par la suite:

Nombre de noeuds du réseau physique	= 40.
Nombre d'arcs non orientés	= 62.
Degré moyen de connexité	= 2 à 3.
Longueurs des câbles	= 5 km.
Longueurs des arcs	= 5 à 15 km.
Capacité des arcs	= de 1000 à 10000 circuits.
Demande moyenne entre centraux	= 50 circuits.
Variance de la demande	= de 10 à 1000 circuits.
Taux moyen de défaillance	= 2 coupures par années par 1000 km de cables.
Temps moyen de réparation	= 15 heures par coupure.
Autres remarques	= Réseaux à haute capacité au centre et beaucoup moins de capacité en périphérie. Il y a deux centraux logiques par noeud physique.

D'après les renseignements que nous venons de voir, nous avons construit le réseau "RED" montré dans la figure 5.1. "RED" est constitué par 40 noeuds et 62 arcs non orientés. Dans la figure 5.1, nous avons spécifié le numéro de chaque noeud ainsi que la capacité de chaque arc exprimée en milliers de circuits. Les petites lignes verticales qui apparaissent au milieu de chaque arc correspondent à la longueur (en unités de 5 km.) de l'arc en question. Pour ce qui est de la demande, nous avons créé une matrice non symétrique de 40×40 dont les entrées correspondent autant que possible aux données fournies, les demandes entre paires de noeuds au centre-ville étant beaucoup plus forte que celles entre les noeuds en périphérie.

5.2 Approximations de la fonction objectif

Nous avons montré dans le chapitre 4 que si tous les produits du réseau ont la même importance (c.-à-d., même poids dans la fonction objectif), la fonction à minimiser est séparable par arcs. Puisque nous considérons les capacités des arcs comme étant fixées, nous avons toujours exprimé le terme de pénalité $\mathcal{P}(X_a)$ donné par l'expression (4.43) comme une fonction de la demande totale sur les arcs. Cependant il s'agit, en réalité, d'une fonction à deux variables, la capacité et la demande sur l'arc. Il s'ensuit que nous avons une fonction différente à évaluer pour chaque type d'arc (le "type" étant défini par la capacité de l'arc). Le deuxième point à considérer était le comportement asymptotique de la fonction; en effet, un calcul simple montrait que la fonction tend vers ses asymptotes $(\mathcal{P}(X) = 0$ si $X < C, \mathcal{P}(X) = X - C$ si $X \geq C)$ lorsque la capacité de l'arc devient très grande. Ce comportement asymptotique n'étant pas convenable pour un algorithme utilisant la notion de gradient, nous avons décidé d'approximer la fonction en trois parties:

$$\mathcal{P}(X) = \begin{cases} 0 & \text{si } X < C - \epsilon, \\ AX^2 + BX + \xi & \text{si } C - \epsilon \leq X < C + \epsilon, \\ X - C & \text{si } X \geq C + \epsilon. \end{cases} \tag{5.1}$$

Pour que $\mathcal{P}'(X)$ soit continue, nous avons:

$$A = \frac{1}{4\epsilon} \quad ; \quad B = \frac{\epsilon - C}{2\epsilon} \quad ; \quad \xi = \frac{(C-\epsilon)^2}{4\epsilon} \tag{5.2}$$

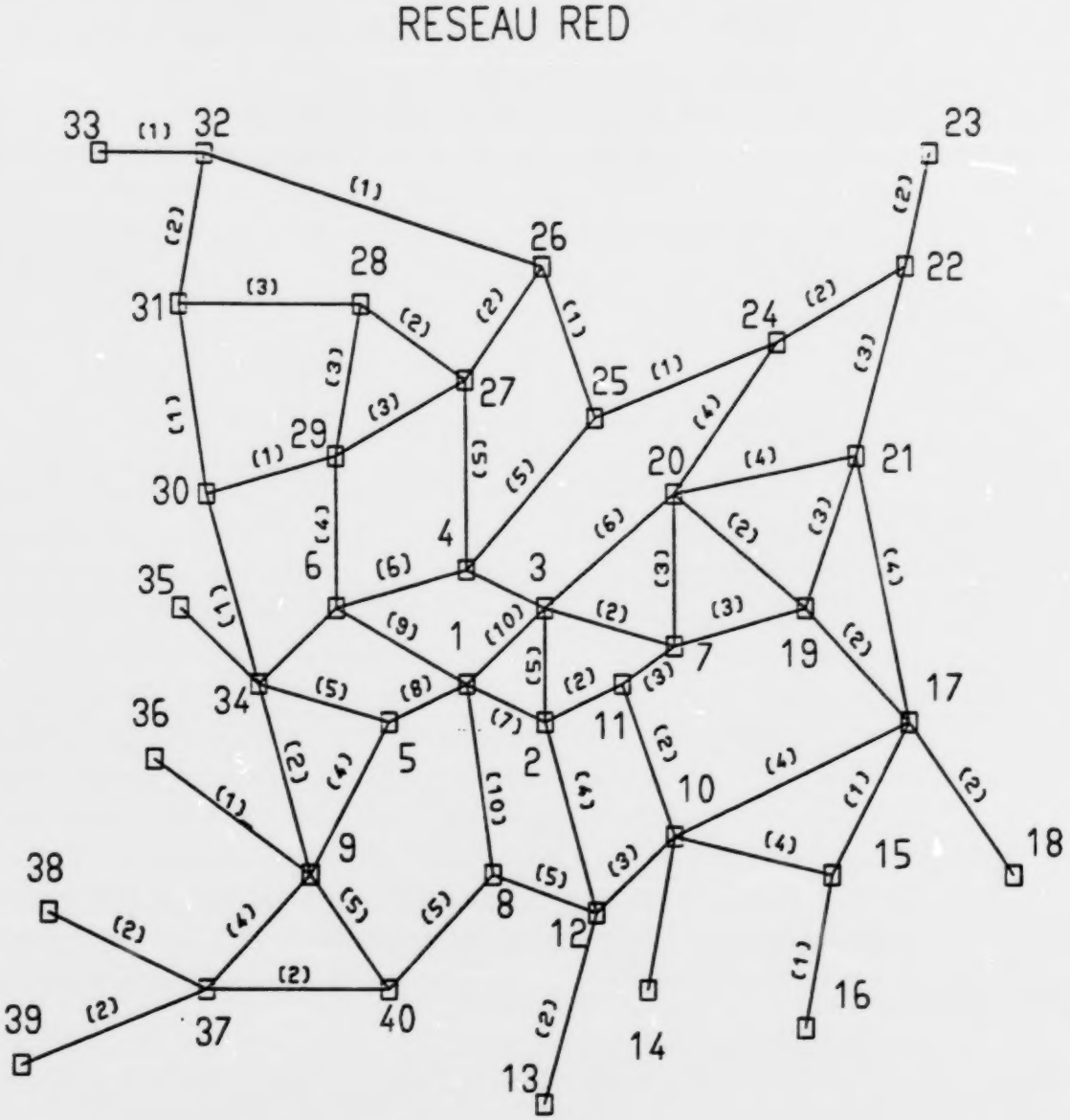


Figure 5.1 Réseau "RED"

La valeur de ϵ dans (5.2) varie en fonction de la capacité sur les arcs du réseau. Afin de choisir une bonne approximation, il fallait d'abord connaître la vraie valeur $\mathcal{P}(X) \forall X$. L'ordre de grandeur des capacités du réseau RED étant dans les milliers de circuits, nous avons recouru à un artifice informatique pour pouvoir évaluer la fonction $\mathcal{P}(X)$ sans passer par le calcul direct des factorielles. Il s'agit de ré-écrire l'inverse de la fonction $\mathcal{P}(X)$ sous la forme de l'équation (4.44):

$$\mathcal{F}(X) = \frac{1}{\mathcal{P}(X)} = \sum_{t=0}^{C} \frac{C!}{(C-t)!} \frac{1}{X^{t+1}} \tag{4.44}$$

et deuxièmement reformuler (4.44) comme suit:

$$\mathcal{F}(X) = \sum_{j=0}^{C} A_j \tag{5.3}$$

A_j étant définie comme

$$A_j = \begin{cases} \frac{1}{X} & \text{si } j = 0, \\ \left[\frac{C-(j-1)}{X}\right] A_{j-1} & \text{si } j = 1, \ldots C. \end{cases} \tag{5.4}$$

L'équation précédente nous permet d'évaluer $\mathcal{F}(X)$ au moyen d'une boucle qui effectue C fois un simple produit.

Une fois que nous connaissons les vraies valeurs de la fonction $\mathcal{P}(X)$, nous avons choisi le ϵ comme étant

$$\epsilon = \alpha(C) \cdot C \tag{5.5}$$

où

$$\alpha(C) = -6.666666 \times 10^{-8} C + 0.086666666. \tag{5.6}$$

Nous allons maintenant justifier l'utilisation de l'expression (5.6). Nous avons déjà mentionné que la fonction de débordement $\mathcal{P}(X)$ avait un comportement de plus en plus "carré" (tendant vers ses asymptotes) au fur et à mesure que la capacité de l'arc augmente. L'idée de l'approximation étant de travailler avec

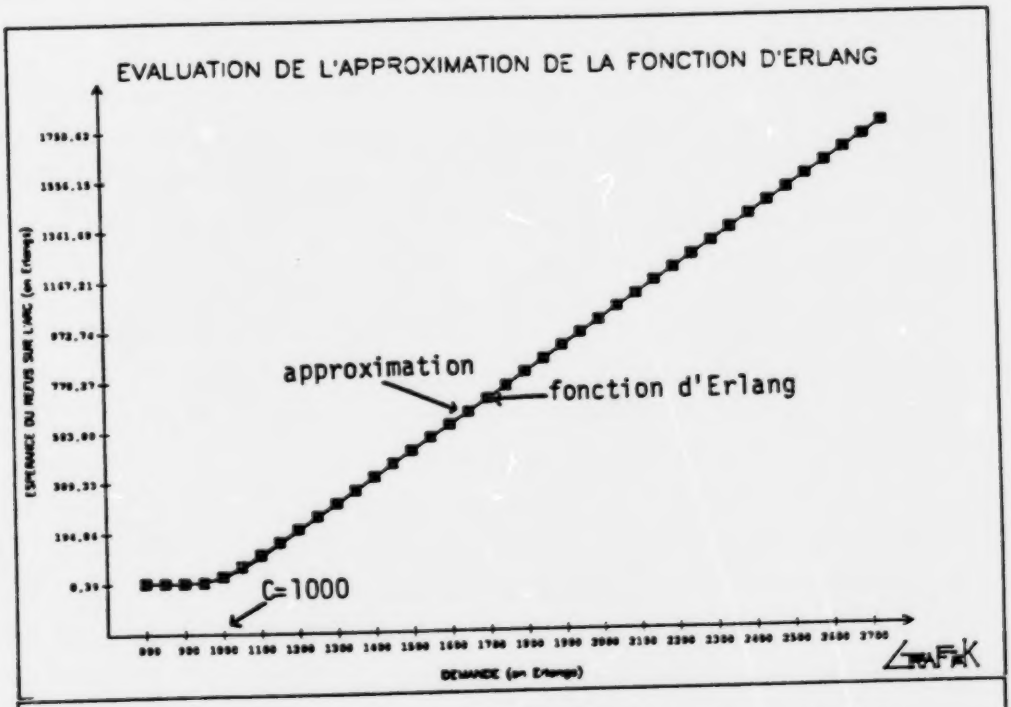
une fonction moins "carrée" et donc plus facile à optimiser avec l'algorithme de Frank-Wolfe, nous voulions quand même que la fonction employée ait le même comportement que la fonction originale. Les expressions (5.5) et (5.6) à utiliser dans l'approximation représentent un compromis numérique entre le comportement naturel de la fonction de débordement et la nécessité de minimiser une fonction qui ne soit pas trop "carrée". La figure 5.2 montre des courbes illustrant la qualité de l'approximation pour des arcs de 1000 et 7000 circuits de capacité.

5.3 Résultats du routage

Avant de présenter les résultats obtenus en routage, il est important de faire une remarque à propos de la terminologie utilisée. Le terme "refus", qui sera amplement employé par la suite, est équivalent aux termes "trafic bloqué", "trafic perdu" ou "trafic de débordement". Nous allons, maintenant, présenter la fonction objectif que, en principe, nous avons choisi de minimiser. Il s'agit de la fonction définie par l'expression (4.38):

$$\sum_{u \in M} \Big(K\mathcal{L}(X_u) + \mathcal{P}(X_u) + M\mathcal{P}^2(X_u) \Big) \tag{4.38}$$

où $\mathcal{L}(X_u)$ est un terme de plus courts chemins, $\mathcal{P}(X_u)$ correspond à la fonction de refus par arc pour chaque arc du réseau, K et M sont des poids que l'on assigne au terme de plus court chemin et au terme de pénalité quadratique. Pour ce qui est de la matrice de demande, nous avons, tel que mentionné dans la première section, créé des données obéissant à certaines caractéristiques. Une fois la demande de base établie, nous avons évalué l'algorithme en variant cette matrice de demande par une constante de proportionalité α. L'algorithme de Frank-Wolfe gère les flots totaux par arc. Puisqu'il était intéressant d'évaluer le refus par origine et même par origine-destination, nous avons créé deux modifications au programme de base qui permettent d'obtenir plus de renseignements à propos des flots par type de produit.



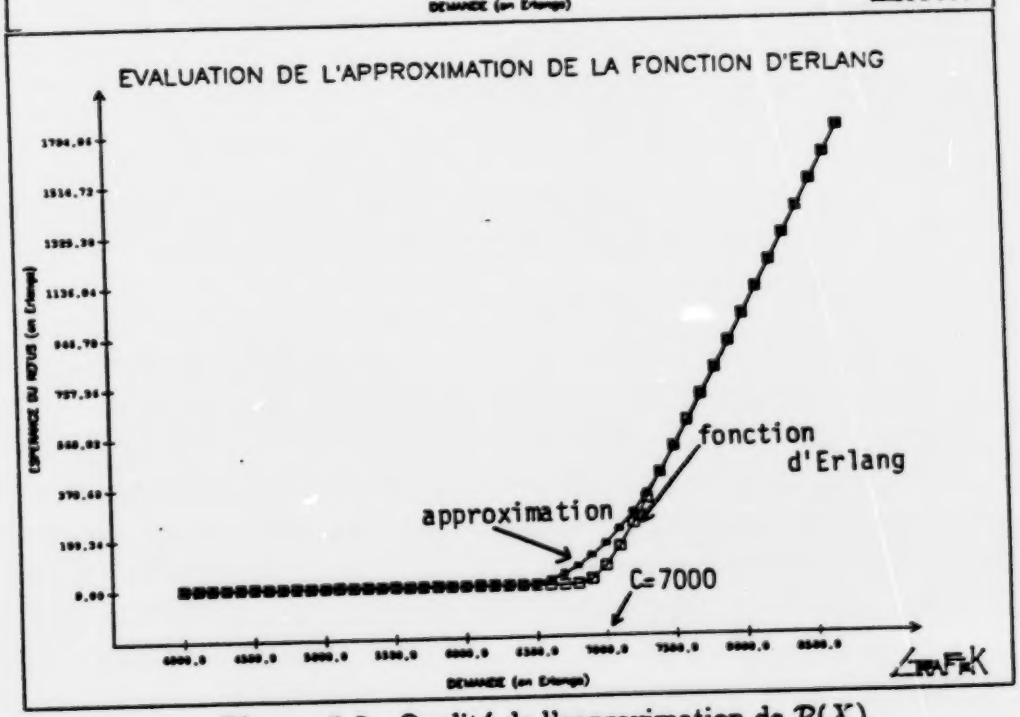


Figure 5.2 Qualité de l'approximation de $\mathcal{P}(X)$.

La figure 5.3 montre l'ensemble d'options pouvant être choisies dans nos programmes de routage.

Nous allons, avant tout, montrer une sortie du programme de base. La sortie, qui correspond à la solution du problème de la minimisation de l'expression (4.38) avec $K = 0$, $M = 0$ et un critère d'arrêt de 1%, est montré dans le tableau 5.1. Le critère d'arrêt étant définit en fonction du pourcentage de diminution de l'objectif par rapport à l'itération précédente.

Le programme dont les résultats correspondent au tableau 5.1 a été écrit en PASCAL et implanté dans une machine SUN 3/50. La solution du problème de base de routage qui calcule, en plus des flots totaux sur les arcs, les flots par origine par arc, a été trouvée en 57 sec. de temps de CPU. Si ce chiffre semble élevé, il faut prendre en considération que l'ensemble des programmes créés constitue un simple prototype de ce qui serait un logiciel de routage à implanter dans une compagnie de télécommunications. Plus d'effort a été mis sur la modélisation du problème de routage que sur le développement et les raffinements informatiques. Malgré ce que l'on vient de mentionner, nous pensons que la performance de l'algorithme est de toutes façons comparable à certaines implantations de l'algorithme de Frank-Wolfe ou du simplexe convexe qui ont été faites pour des problèmes de taille équivalente à RED.

Nous montrons, dans le tableau 5.2, les résultats obtenus par Guelat (1983) et Sauvé (1984) pour le problème d'affectation du trafic d'équilibre de la ville de Hull.

Le tableau 5.2 montre que le problème de HULL est, en termes de variables, 2.5 fois plus grand que RED et, en termes de contraintes, 4.2 fois plus grand. Cependant, le nombre de produits que RED manipule est plus que deux fois supérieur au nombre de produits du réseau de HULL et, puisque la complexité de résolution augmente de façon plus que linéaire avec le nombre de produits du réseau, nous jugeons que HULL et RED sont comparables. Si l'on étudie maintenant les machines utilisées pour les calculs, nous mentionnons que le SUN 3/50 est considéré comme un micro avec une horloge de 16.67 MHz tandis que le CYBER 835 est un ordinateur central.

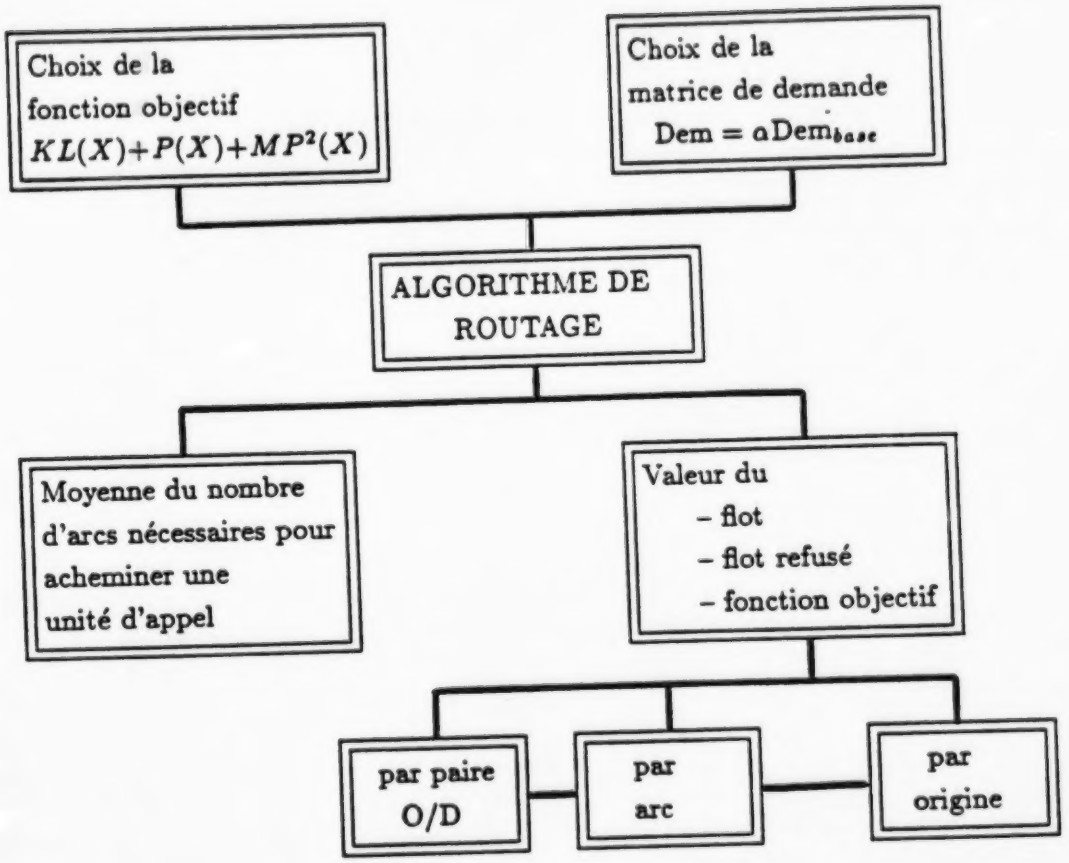


Figure 5.3 Organigramme des résultats pouvant être obtenus par les programmes de routage.

TABLEAU 5.1

Solution au problème de routage pour
$K = 0$, $M = 0$, $\alpha = 1$, 1% de critère d'arrêt.

ARC	REFUS	DEMANDE	CAPACITE	ARC	REFUS	DEMANDE	CAPACITE
(1,2)	139.26	7114.93	7000	(10,17)	219.69	4219.24	4000
(1,3)	878.14	10878.14	10000	(12,13)	760.00	2760.00	2000
(1,5)	287.03	8287.03	8000	(15,16)	11.25	980.00	1000
(1,6)	0.00	8747.06	9000	(15,17)	88.08	1088.08	1000
(1,8)	1463.40	11463.40	10000	(17,18)	0.00	1000.00	2000
(2,3)	18.17	4872.54	5000	(17,19)	2.05	1888.02	2000
(2,11)	170.20	2170.20	2000	(17,21)	48.63	3976.08	4000
(2,12)	150.11	4139.62	4000	(19,20)	0.00	1627.15	2000
(3,4)	0.00	9748.65	10000	(19,21)	0.00	1184.08	3000
(3,7)	168.25	2168.25	2000	(20,21)	0.00	3471.96	4000
(3,20)	158.74	6141.65	6000	(20,24)	0.00	2357.99	4000
(4,6)	2231.43	8231.43	6000	(21,22)	0.00	2159.55	3000
(4,25)	0.00	3093.76	5000	(22,23)	0.00	690.00	2000
(4,27)	299.42	5299.42	5000	(22,24)	0.00	1831.26	2000
(5,9)	834.20	4834.20	4000	(24,25)	1331.93	2331.93	1000
(5,34)	0.00	3684.72	5000	(25,26)	95.32	1095.32	1000
(6,29)	66.38	4012.44	4000	(26,27)	0.00	1837.68	2000
(6,34)	2315.69	6315.69	4000	(26,32)	0.01	921.46	1000
(7,11)	25.52	2942.90	3000	(27,28)	39.07	2004.72	2000
(7,19)	0.00	1932.53	3000	(27,29)	0.00	2718.38	3000
(7,20)	0.00	2712.10	3000	(28,29)	0.00	2534.12	3000
(8,12)	797.62	5797.62	5000	(28,31)	0.00	2191.39	3000
(8,40)	1670.32	6670.32	5000	(29,30)	9.92	976.33	1000
(9,34)	2420.56	4420.56	2000	(30,31)	26.34	1011.81	1000
(9,36)	0.00	830.00	1000	(30,34)	680.13	1680.13	1000
(9,37)	0.00	1492.02	4000	(31,32)	0.00	1670.56	2000
(9,40)	0.00	4452.47	5000	(32,33)	0.00	780.00	1000
(10,11)	35.06	1996.75	2000	(34,35)	0.00	1230.00	2000
(10,12)	1636.19	4636.19	3000	(37,38)	0.00	970.00	2000
(10,14)	370.00	1370.00	1000	(37,39)	0.00	820.00	2000
(10,15)	0.00	2280.52	4000	(37,40)	0.00	1727.98	2000
				total	19448.09	208472.32	

Moyenne du nombre d'arcs
parcourus par unité de flot = 4.72

Selon les utilisateurs des deux machines le CYBER serait environ 5 fois plus rapide que le SUN 3/50. Il s'ensuit que les résultats obtenus pour le réseau RED soient comparables aux temps de calcul existants pour le réseau HULL.

TABLEAU 5.2

Résultats comparatifs des réseaux HULL et RED.

Problème	Noeuds	Arcs	Flots	Variables	Contraintes	Critère d'arrêt	Temps (sec.) CPU		
							(Cyber 835)		(SUN 3/50)
							Guelat (F-W)	Sauvé (S. C.)*	Sanso
HULL	424	798	16	12768	6784	0.1%	75.02	33	
RED	40	124	40	4960	1600	0.1%	—	—	134

* simplexe convexe

Nous allons, maintenant, montrer sous forme de graphiques, les résultats de certains tests réalisés au niveau du routage.

Un paramètre qui influence fortement le temps de calcul est le niveau de demande. La figure 5.4 montre, dans un même graphique, le temps d'exécution vs la proportion de demande pour un programme où les produits sont les appels par origine et pour celui où les produits sont les appels origine–destination. Les valeurs de K et M dans la fonction objectif ont été fixées à zero. La forme des deux courbes est la même mais la figure montre que l'algorithme le plus détaillé est sept fois plus lent que celui qui travaille avec les flots par origine seulement.

Dans la figure 5.5 nous montrons en détail la variation du temps d'exécution vs α lorsque l'algorithme utilisé considère les produits comme étant les flots par origine.

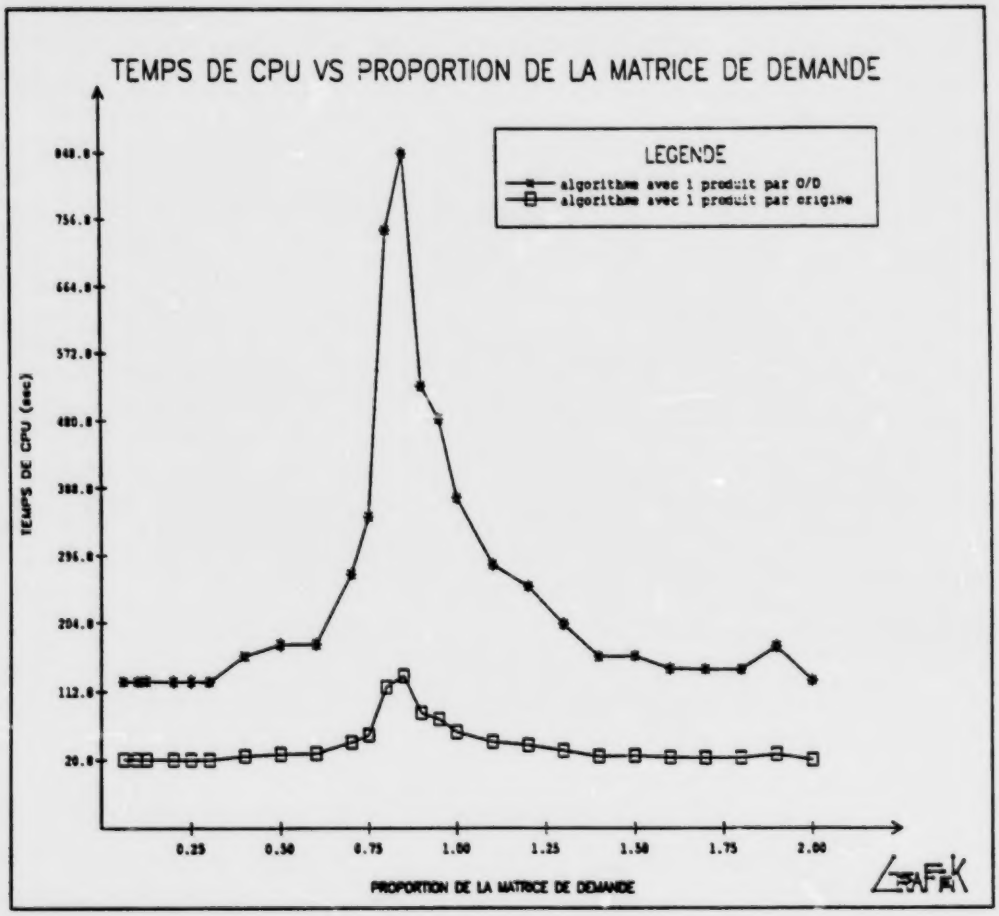


Figure 5.4 Temps d'exécution des programmes de routage vs proportion de la matrice de demande.

L'on remarque que pour des petites et des grandes valeurs de α le temps de CPU est petit et presque constant. Dès que la proportion de demande commence à augmenter, le temps d'exécution augmente aussi et il rejoint son maximum lorsque $\alpha = 0.85$. Ce type de comportement est tout à fait normal pour un algorithme de Frank-Wolfe et nous avons deux façons de l'expliquer. D'abord, intuitivement, lorsque la demande est très faible ou très grande, il est très facile pour l'algorithme

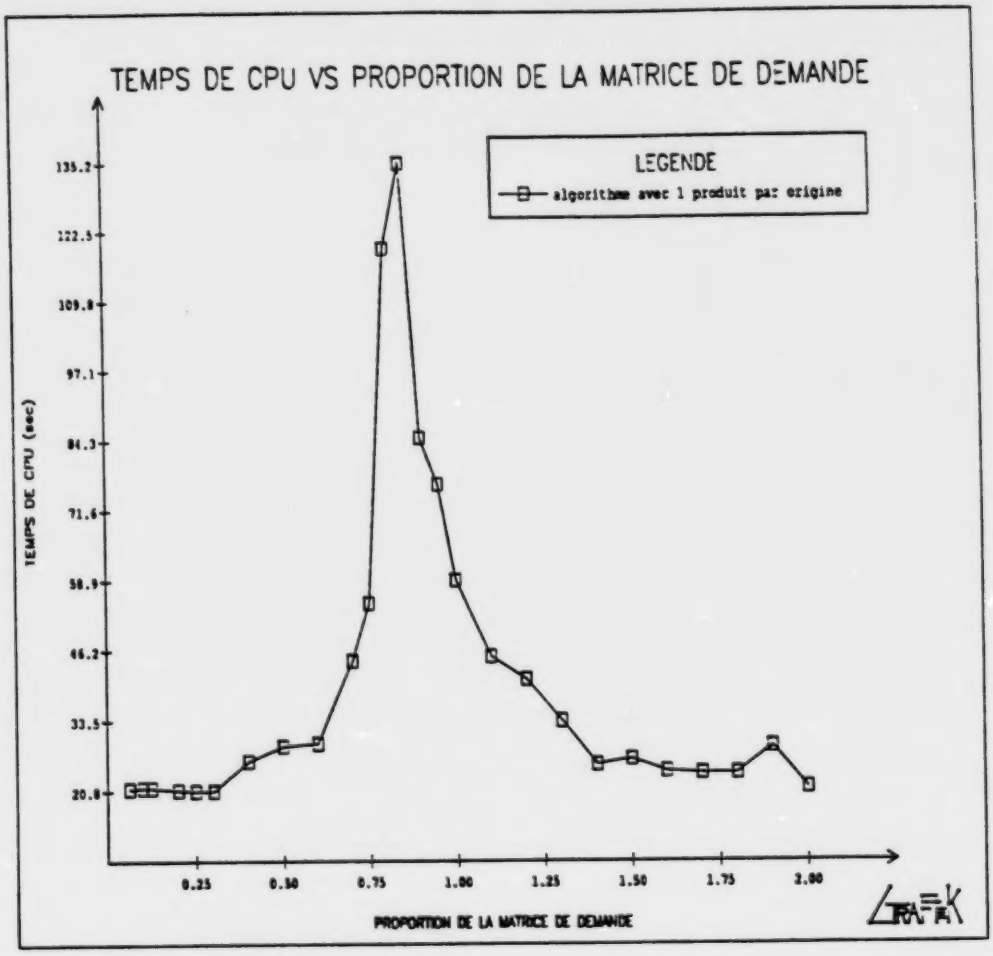


Figure 5.5 Temps d'exécution vs proportion de la matrice de demande pour un algorithme considérant un produit par noeud.

de trouver une solution optimale, par contre, lorsque la demande est grande sans congestionner complètement le réseau, l'algorithme doit réaliser plus d'itérations pour trouver son optimum. La deuxième explication est reliée à la forme de la fonction objectif. Nous avons vu qu'elle est séparable par arcs et que la fonction par arc est linéaire, lorsque la demande est très faible ou très forte, et quadratique dans un intervalle qui inclut le point de demande égal à la capacité de l'arc. Il s'ensuit que la fonction globale résultante est aussi linéaire aux extrêmes et très

non-linéaire au milieu. Le Frank-Wolfe, alors, donnera une solution optimale en une itération lorsque la fonction objectif est linéaire et nécessitera de plus en plus d'itérations au fur et à mesure que la fonction devient non-linéaire.

Jusqu'ici nous avons considéré que K et M dans la fonction objectif sont nuls. Nous voulons maintenant étudier le comportement du trafic d'appels lorsque l'on fait varier ces paramètres.

La figure 5.6 indique la moyenne du nombre d'arcs parcourus par unité de flot lorsque l'on fait varier K, c.-à-d. le poids du terme de plus court chemin dans la fonction objectif. Nous avons réalisé le test pour plusieurs proportions de la matrice de demande.

Nous observons que les trois courbes de la figure 5.6 ont un comportement similaire; la moyenne d'arcs parcourus par une unité de flot est fortement réduite lorsque l'on introduit le terme de plus court chemin dans l'objectif, une réduction est encore possible en augmentant le poids K du terme mentionné. Cependant, après avoir atteint la valeur $K = 1.5$, il n'est plus possible de réduire le nombre d'arcs qu'une unité de flot doit parcourir en moyenne. Du côté de la valeur de α, le comportement des courbes de la figure 5.6 confirme les résultats montrés dans les figures 5.4 et 5.5. Lorsque la fonction objectif est très non-linéaire ($\alpha = 0.8$), le nombre moyen d'arcs parcourus par unité de flot est plus élevé car le flot pour une paire O/D est réparti sur plusieurs chemins et certains de ces chemins sont plus longs, d'un autre côté l'effet "bénéfique" du terme linéaire de plus court chemin est plus important pour ce cas. Le cas $\alpha = 2$ montre que lorsque le réseau est très congestionné, la fonction objectif devient presque linéaire, la plupart de la demande est refusée et le flot accepté est acheminé par le plus court chemin de façon naturelle. Ce qui explique que l'introduction d'un terme de plus court chemin ne cause pas d'effet aussi important que pour le réseau à moindre congestion.

Nous passons, maintenant, à l'étude de l'effet de la variation du poids M de la fonction objectif. L'utilité du paramètre M et du terme quadratique est d'équilibrer les pertes d'appels par origine. Théoriquement, ce comportement bénéfique du routage pourrait être atteint en introduisant un terme de variance

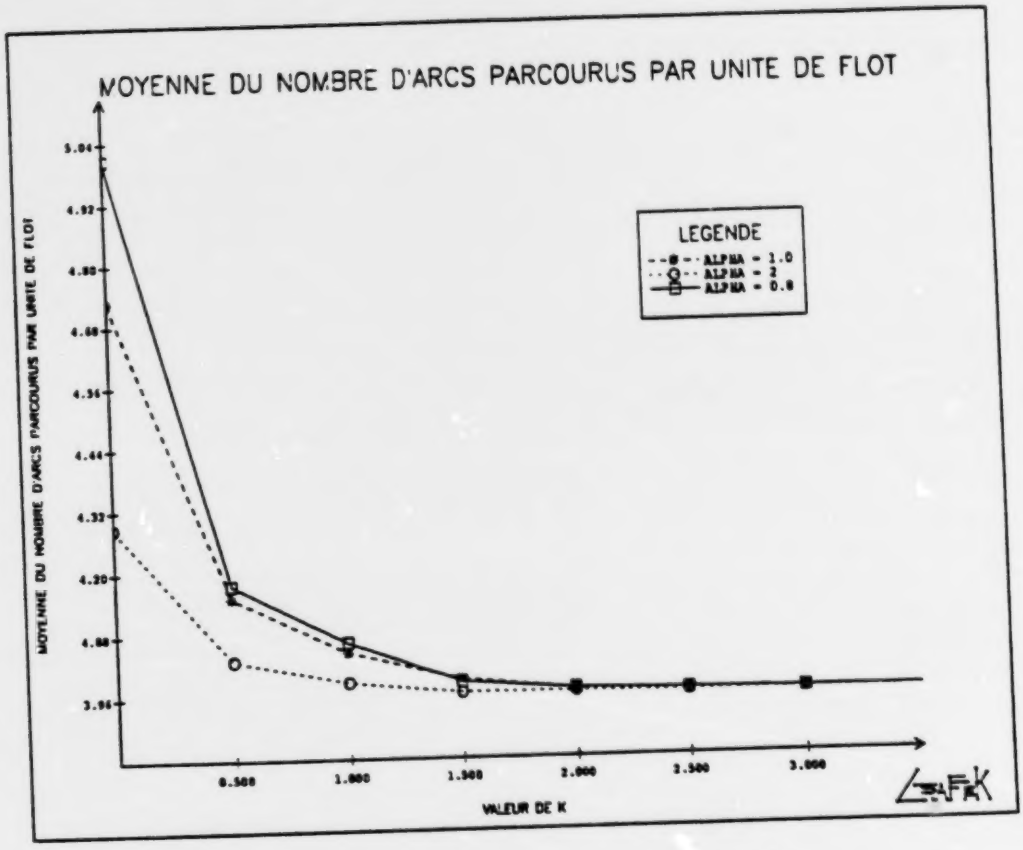


Figure 5.6 Moyenne du nombre d'arcs parcourus par unité de flot.

du refus parmi tous les noeuds origine du réseau. Notre algorithme travaillant avec une fonction objectif séparable par arc, il n'était pas possible d'ajouter un tel terme à la fonction objectif. Nous avons donc introduit le terme quadratique de pénalisation du refus sur chaque arc en espérant qu'il rende le refus plus uniforme dans l'ensemble du réseau et que donc, indirectement, il produise un effet favorable dans le calcul du refus par origine.

Le terme $\sum_a MP^2(X_a)$ montré dans l'expression (4.38) n'est pas, cependant, le plus convenable à ce type de test car l'on risque d'avoir une situation où l'uniformité du refus sur les arcs est atteinte en soulageant des arcs à capacité

élevée au détriment des arcs à faible capacité. Cette situation est corrigée en utilisant un terme qui prend en considération la capacité des arcs. La fonction objectif choisie pour le test de M a donc été:

$$\sum_{u} \left(\mathcal{P}(X_u) + M(\mathcal{P}(X_u)/C_u)^2 \right) \tag{5.7}$$

Nous avons étudié la variation de deux paramètres du routage: le refus moyen par arc et la variance du % de refus sur tous les arcs du réseau; le refus et le % de refus étant définis comme suit:

$$\text{refus sur l'arc } u = \mathcal{P}(X_u) \tag{5.8}$$

et

$$\%\text{ de refus sur l'arc } u = \frac{\mathcal{P}(X_u)}{C_u} \times 100 \tag{5.9}$$

Les figures 5.7 et 5.8 montrent, respectivement, la variation des paramètres mentionnés par rapport au poids M.

Nous constatons que l'augmentation de M fait diminuer la variance du % de refus de façon presque monotone. La moyenne du refus, par contre, augmente jusqu'à $M = 25$ ce qui constitue, à peu près, le seuil d'équivalence de $\mathcal{P}(X)$ et $M(\mathcal{P}(X)/C)^2$. À partir de $M = 25$, le terme quadratique devient beaucoup plus important, ce qui crée une distorsion de la fonction objectif. Pour ce qui est du pas ΔM, il a été choisi de façon à ce qu'il n'interfère pas avec le critère d'arrêt de l'algorithme.

En ce qui concerne l'hypothèse d'avoir une meilleure répartition du % de refus par origine lorsque le % de refus par arc est mieux réparti, nous mentionnons qu'elle s'est avérée fausse. Nous mentionnons deux explications à ce phénomène: l'effet de la structure du réseau et la non-unicité de la solution par produit.

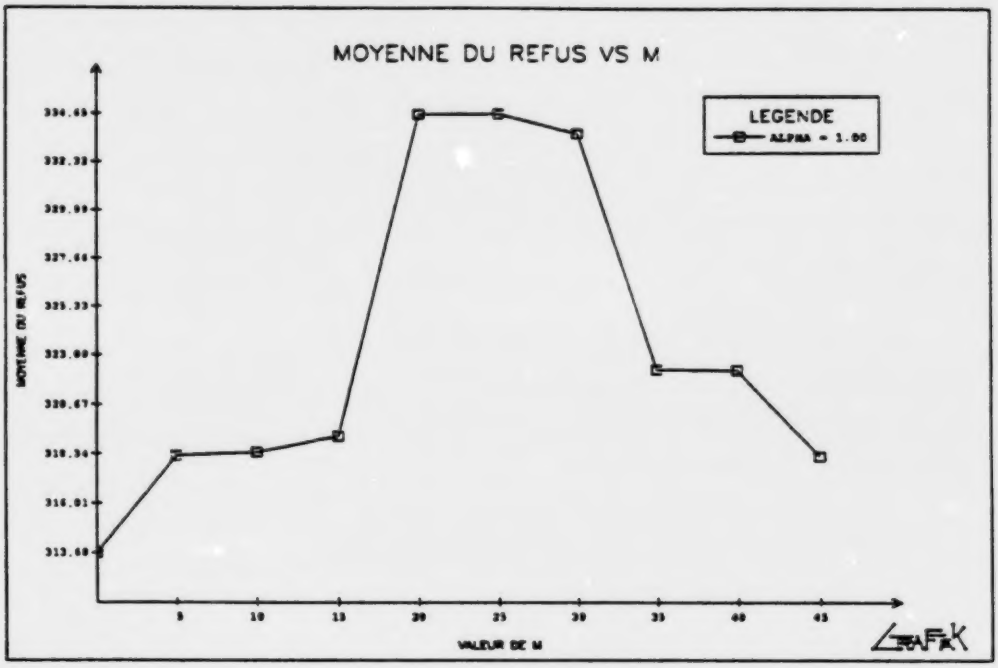


Figure 5.7 Moyenne du refus vs M.

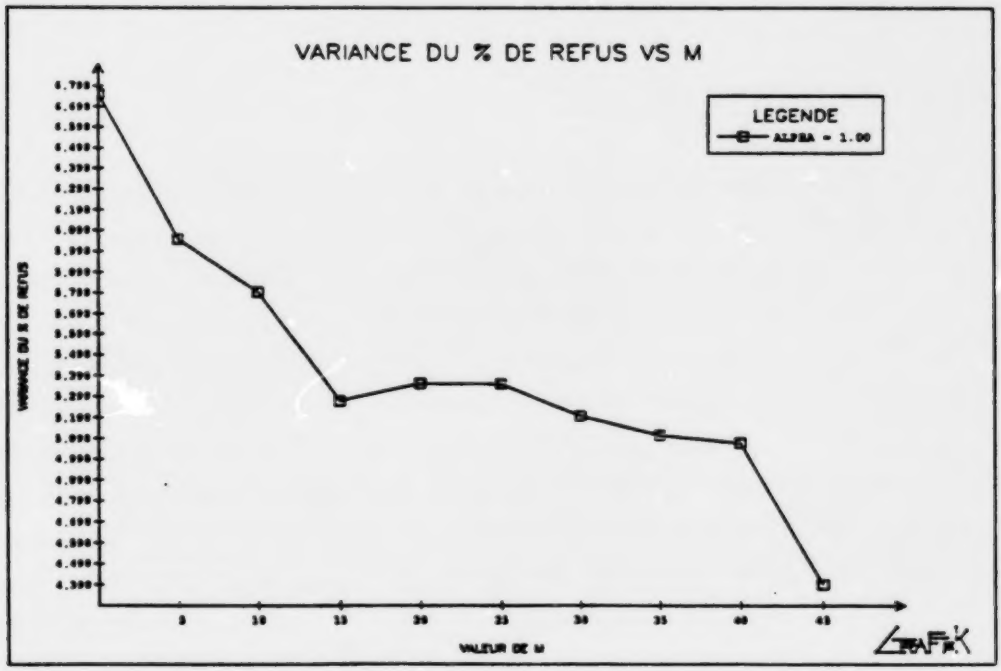


Figure 5.8 Variance du % de refus vs M.

Étudions d'abord la structure du réseau. Nous avons construit, à partir de caractéristiques réelles, un réseau à haute capacité au centre et faible capacité en périphérie. Les demandes par paire origine-destination répondent aussi à ce type de comportement. Il s'ensuit que même si dans l'ensemble du réseau la variance du % de refus par arc a diminué, la structure des chemins qui relient certains noeuds origine-destination est telle qu'il est impossible de réduire leur % de refus par produit.

Le deuxième point mentionné est le fait que l'algorithme converge vers une solution unique par arc, cependant il existe plusieurs combinaisons de solutions par produit qui donnent la même solution par arc. Nous donnons dans la figure 5.9 un exemple de ce comportement: les demandes de A et B vers le noeud E sont 10 et 5 unités respectivement. Les valeurs entre parenthèses indiquent la solution optimale par arc, les paires ordonnées entre les crochets indiquent des solutions possibles par produit $[X^A, X^B]$.

Il s'ensuit que des facteurs tel que la façon de programmer l'algorithme de Frank-Wolfe ou la solution de départ peuvent faire changer notablement la solution optimale par produit.

5.4 Résultats en fiabilité

Nous avons mentionné dans le chapitre 3 qu'une bonne mesure de fiabilité était l'espérance du trafic perdu dû aux pannes. Pour calculer cette mesure, nous avons suggéré l'utilisation des bornes de Li et Silvester. Une fois les bornes supérieures et inférieures du refus total calculés, nous proposions de soustraire de ces dernières valeurs le refus résultant de l'état de fonctionnement parfait, c.-à-d. le refus dû à un manque de capacité dans le réseau. Puisque c'était notre but au départ, nous allons, plus tard dans la section, montrer les résultats du refus dû uniquement aux pannes. Cependant, nous jugeons que, étant donné que les réseaux de télécommunications existants sont parfois sur-utilisés, à cause, par exemple, de l'accroissement imprévu de la demande, il est intéressant d'étudier le flot non acheminé total dans le réseau, c.-à-d. celui qui est dû à des problèmes de

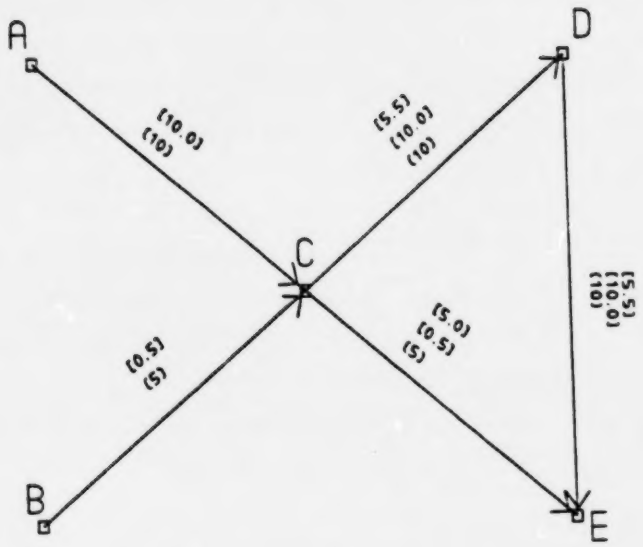


Figure 5.9 Exemple de la non-unicité d'une solution par produit.

manque de capacité. Nous allons voir que la même procédure utilisée pour évaluer des bornes sur l'espérance du refus total d'un réseau existant peut être employée pour la conception de réseaux.

Le premier obstacle que nous avons rencontré pour l'évaluation des bornes du refus avec la méthodologie proposée était la mauvaise performance de l'algorithme de Frank-Wolfe au niveau de la réoptimisation. L'idée de départ de notre méthode est basée sur le fait que lorsque un arc tombe en panne, la répartition des flots dans le réseau est très semblable à celle que l'on avait lorsque le réseau fonctionne parfaitement. Or il était justifié, lorsque l'on considérait les états de panne, de partir de la solution optimale du routage de l'état de fonctionnement parfait et d'ajouter un grand coût de pénalité au flot sur l'arc en panne. Cette procédure obligeait l'algorithme à enlever du flot sur l'arc en panne. Malgré le fait que l'algorithme de Frank-Wolfe est reconnu pour ne pas être très efficace pour soustraire du flot, on ne s'attendait pas à de si mauvaises performances: le nombre d'itérations pour réoptimiser était, pour certains états, de deux à trois fois le nombre d'itérations nécessaires pour atteindre le minimum lors de la première

optimisation. Il était donc plus convenable d'optimiser chaque état à partir de la même solution réalisable de départ que d'essayer de réoptimiser à partir de la solution optimale de l'état de fonctionnement. Le temps de calcul des 63 états étudiés était alors d'environ 63 fois le temps de calcul du routage pour l'état de fonctionnement parfait; nous devions donc allouer une heure à l'obtention de nos résultats.

Le facteur temps nous a forcé à reformuler le problème en termes de précision: étant donné les faibles probabilités de pannes des réseaux existants, était-il si nécessaire de demander autant de précision aux résultats de l'état de fonctionnement qu'à ceux des états de panne? Un calcul rapide nous a permis de répondre négativement à cette question et les résultats de plusieurs tests que nous montrerons par la suite nous ont donné raison.

Nous avons abordé le problème de la façon suivante: nous avons donné deux critères d'arrêt différents à l'algorithme de Frank-Wolfe; un, très strict, pour l'état de fonctionnement parfait et un autre, beaucoup plus large, pour les états de panne. La solution de départ ainsi que le nombre d'itérations nécessaires lorsque la fonction objectif est à l'intérieur du critère d'arrêt sont les mêmes pour tous les états. La figure 5.10 montre les bornes supérieures et inférieures de l'espérance du refus total dans le réseau pour quatre cas différents. La paire (crit1, crit2) indique le critère d'arrêt, en pourcentage, donné aux états de fonctionnement et de panne, respectivement.

Avant de continuer avec les détails de la figure mentionnée, nous faisons une parenthèse pour introduire la notion de "cas optimal". Ré-écrivons d'abord l'équation qui correspond à la borne supérieure du refus (eq. 2.6.3) en mettant en évidence l'état de fonctionnement parfait (S_1) et les états de panne (S_k , $k = 2, \ldots, 2^n$):

$$E[Z]_{sup} = P(S_1)\, Z(S_1) + \sum_{k=2}^{m} P(S_k)\, Z(S_k) + \\ + \left(1 - \sum_{k=1}^{m} P(S_k)\right) Z(S_{2^n}). \qquad (5.10)$$

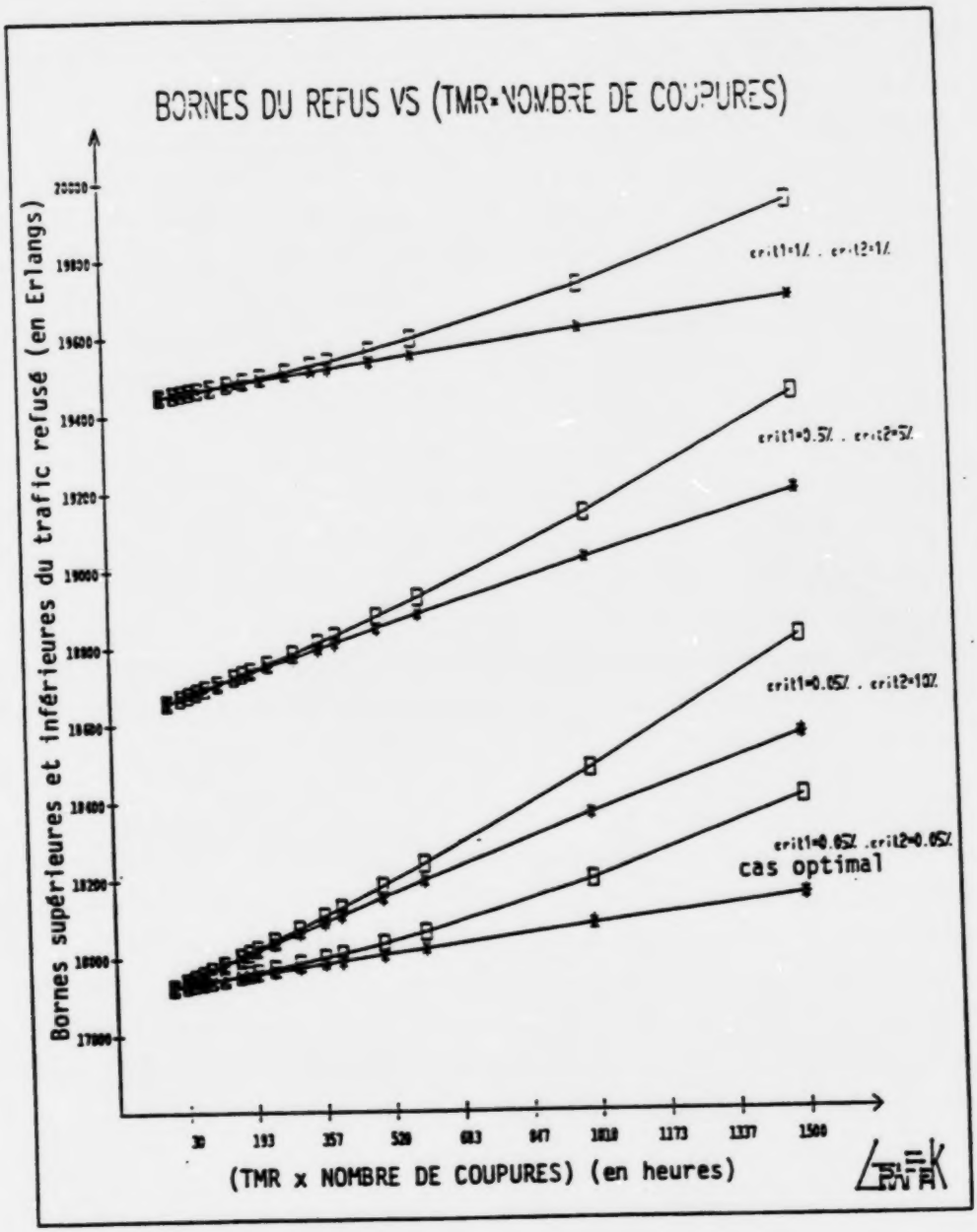


Figure 5.10 Bornes du refus vs temps de défaillance.

L'équation (5.10) est une borne supérieure de l'espérance du refus total dans le réseau lorsque le refus pour le $k^{ième}$ état est donné par la variable $Z(S_k)$. Idéalement, la valeur de $Z(S_k)$ utilisée dans le calcul des bornes devrait être le refus minimal $Z^*(S_k)$ obtenu au moyen de l'algorithme de routage. Or la valeur optimale varie en fonction du critère d'arrêt choisi. Il s'ensuit que les bornes de l'espérance du refus varient aussi en fonction des critères d'arrêt d'où la nécessité de choisir un cas avec des critères d'arrêt assez stricts pour qu'il puisse être considéré comme "minimal". Il s'agit du cas où les deux critères d'arrêt sont 0.05%.

De retour à la figure 5.10, nous montrons la variation des bornes supérieures et inférieures de l'espérance du refus total dans le réseau en fonction du produit temps moyen de réparation par nombre de coupures par année. Ce dernier paramètre constitue le temps moyen de défaillance d'une section d'arc par année, une section étant définie comme 5 Km de câble. Selon les données fournies au début du chapitre, le temps moyen de défaillance d'une section d'arc est de 30 heures par année. Cette valeur est extraite d'une année tout à fait normale. Or nous avons voulu considérer la possibilité de grèves ou de coupures de budget qui de façon indirecte ou directe pourraient augmenter le temps moyen de réparation ou le nombre de coupures par année. Il s'ensuit que la plage de valeurs choisie est de 30 à 1500 heures.

Pour ce qui est du comportement des bornes supérieures et inférieures pour chacun des cas, nous remarquons que les deux bornes sont très rapprochées lorsque le temps de défaillance est petit. Cependant, lorsque ce dernier paramètre croît, la distance entre les bornes est beaucoup plus grande, ce qui fait qu'il est plus difficile d'estimer la valeur du refus total dans le réseau. Cependant, on constate, pour tous les cas étudiés, que la distance entre les bornes supérieures et inférieures est presque la même et quelle croît avec le temps de défaillance. En effet, la différence entre les bornes est donnée par l'expression suivante:

$$\Delta E[Z] = \left(1 - \sum_{k=1}^{m} P(S_k)\right)\left[Z(S_{2^n}) - Z(S_1)\right]. \qquad (5.11)$$

Dans (5.11), $Z(S_{2^n})$ équivaut au refus lorsque tous les arcs sont en panne, c'est-à-dire à la demande totale dans le réseau. $Z(S_1)$, par contre, correspond au refus de l'état de fonctionnement parfait qui, lui, varie selon le critère d'arrêt choisi. Cependant, même pour un temps de défaillance de 1500 heures, la probabilité d'être dans un état de panne double ou plus (représenté par le premier terme de 5.11) est tellement faible que l'on peut considérer que la différence entre les bornes ne varie pas avec le critère d'arrêt.

Par contre, le critère d'arrêt a une influence dramatique sur la valeur absolue des bornes sur l'axe des ordonnées, l'espérance du refus étant beaucoup plus basse pour des petites valeurs du premier critère d'arrêt. Ce comportement est dû au fait, déjà mentionné, que la probabilité de bon fonctionnement du réseau est très élevée, d'où une grande précision dans le calcul du refus pour cet état à une plus grande influence dans l'évaluation de l'espérance du refus.

Si ce dernier résultat semble confirmer notre choix d'augmenter le critère d'arrêt des états de panne tout en diminuant celui de l'état de fonctionnement, la prochaine figure donne des résultats encore plus catégoriques envers la décision choisie.

La figure 5.11 montre la variation du % d'erreur de la borne supérieure du refus en fonction du temps moyen de défaillance par année. L'erreur a été calculée par rapport à la borne supérieure du cas considéré "optimal". Trois cas ont été étudiés: le cas où les deux critères d'arrêt sont de 1%, celui où le critère d'arrêt pour le premier état est de 0.5% et pour les états de panne de 5% et, finalement le cas qui correspond à la paire (0.05%, 10%). Nous constations que non seulement l'erreur introduite par le choix du critère d'arrêt est significative mais qu'il est définitivement convenable, du point de vue de la précision et du temps de calcul, de réduire le premier critère tout en relâchant le deuxième. En effet, pour le cas (1%, 1%), nous avions besoin de 3678.7 sec. pour obtenir un résultat et ce chiffre

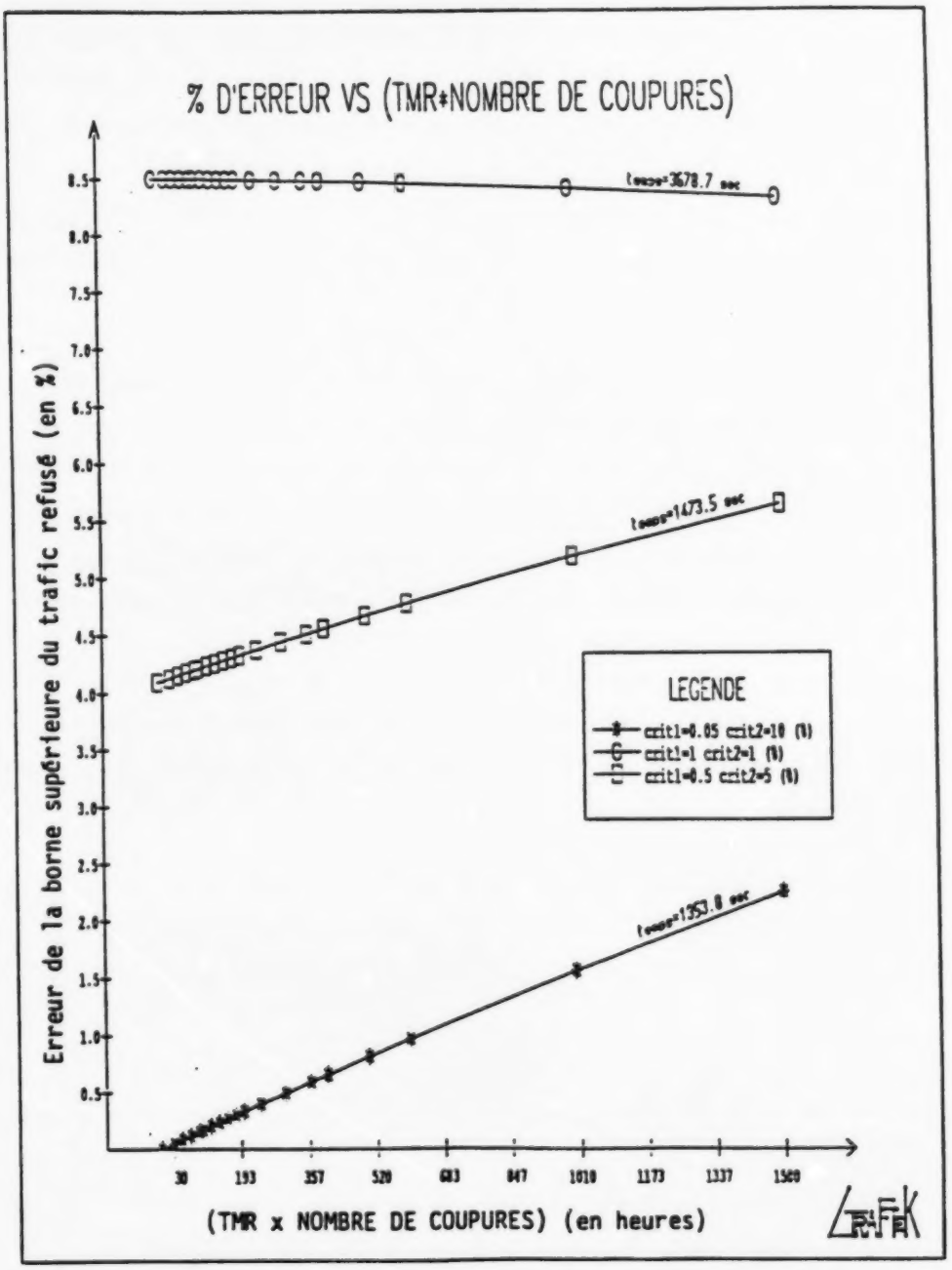


Figure 5.11 Erreur de la borne supérieure du refus vs temps de défaillance.

a presque été divisé par trois tout en augmentant la précision de la réponse de l'algorithme. Ce phénomène de plus de précision à un moindre coût, si rare dans la nature, est dû et à la faible probabilité de panne et au fait que l'on mesure le refus total dans le réseau et non le refus qui provient exclusivement des pannes.

En effet, tel que montré dans la figure 5.12, même si les probabilités de pannes du réseau sont faibles, le fait de soustraire le refus de l'état de fonctionnement parfait crée une grosse erreur dans le résultat voulu si l'on utilise, encore une fois, la procédure des deux critères d'arrêt très distincts.

Dans la figure 5.12, nous observons que la borne supérieure du refus dû aux pannes du cas (1%, 1%) est très proche de celle du cas "optimal". Par contre, lorsque l'on serre le premier critère et que l'on relâche le deuxième, l'erreur par rapport à la borne optimale croît de plus en plus. Ce comportement s'explique par le fait que l'on soustrait de l'expression de borne totale la valeur Z_1, c'est-à-dire la valeur qui est calculée de la façon la plus précise. Or les valeurs des termes qui restent dans l'évaluation des bornes correspondent à celles calculées avec beaucoup plus d'imprécision. Il s'ensuit que le calcul des bornes du refus dû uniquement aux pannes nécessite deux critères d'arrêt serrés et, donc, un grand temps de calcul.

Nous jugeons que, jusqu'à ce point, l'objectif principal du présent travail a été accompli. Cependant, nous nous sommes posé trois questions qui découlent de l'analyse de la performance de la méthode de calcul de fiabilité montrée dans cette section. Les questions sont explicitées par la suite.

a) Est-il possible d'améliorer les bornes du refus (spécifiquement la borne supérieure) sans changer d'algorithme et sans trop augmenter le temps d'exécution?

b) Est-il possible de retrouver une borne supérieure du refus, comparable à celle obtenue, en moins de temps?

c) Comment utiliser la méthode proposée dans la conception de réseau?

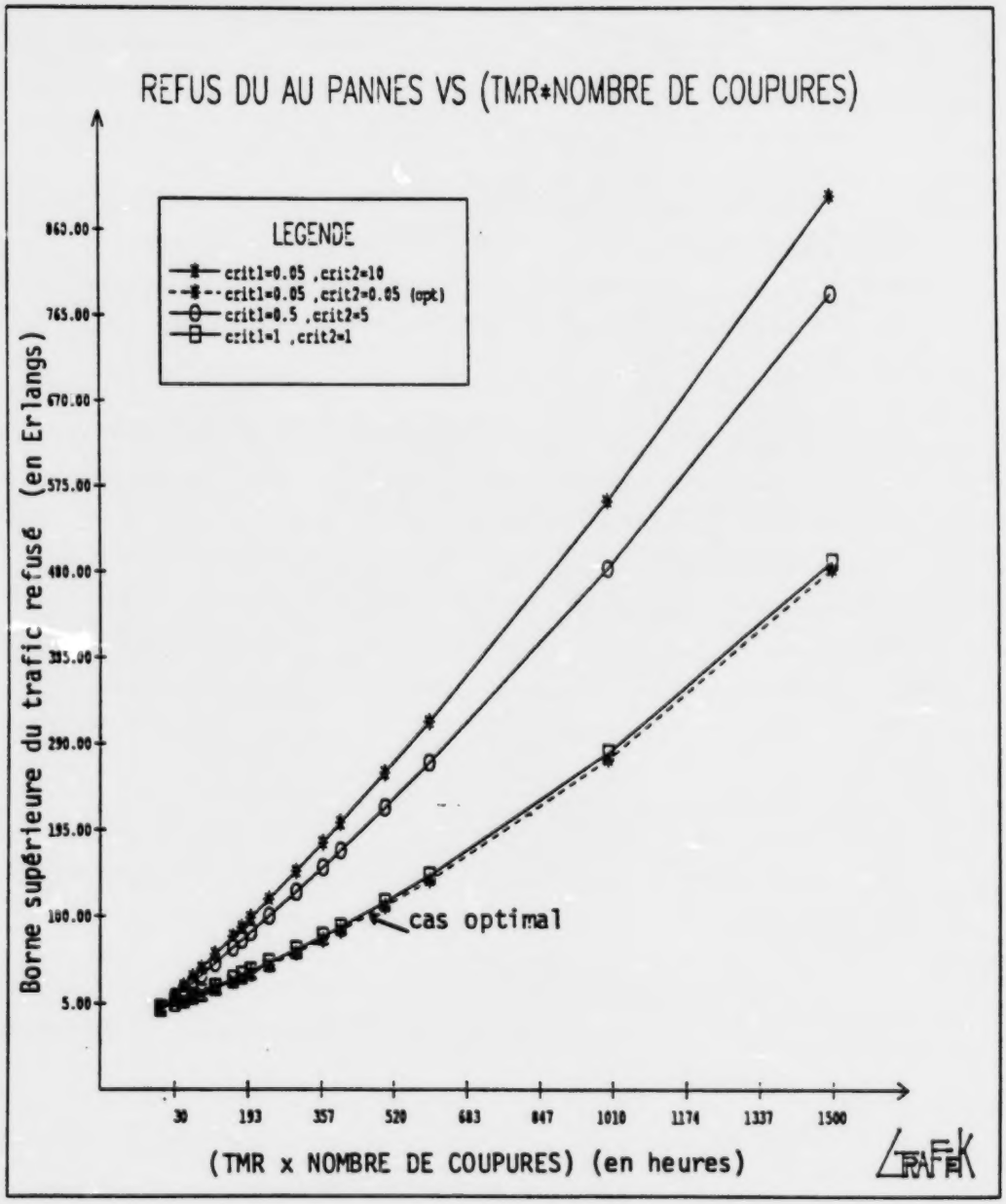


Figure 5.12 Refus dû aux pannes vs temps de défaillances.

Les questions (a) et (b) constituent des ambitions naturelles d'amélioration de la méthode proposée. Cependant, il n'est pas évident qu'il existe un lien entre les deux premières questions et la troisième. Nous incluons une dernière sous-section à ce chapitre afin d'esquisser des réponses aux questions proposées et de montrer le lien entre les trois questions.

5.4.1 Amélioration des bornes et conception de réseaux.

Réécrivons avant tout l'expression de la borne supérieure du refus donné par l'équation (5.10):

$$E[Z]_{sup} = P(S_1)\,Z(S_1) + \sum_{k=2}^{m} P(S_k)Z(S_k) + \\ + \sum_{k=m+1}^{2^n} P(S_k)\cdot Z(S_{2^n}). \qquad (5.12)$$

Le premier terme correspond au produit $Z(S_k)P(S_k)$ lorsqu'il n'y a pas de panne; le deuxième équivaut à la somme de ce produit pour les états de panne simple du réseau. Le troisième terme, cependant, implique que lorsque deux arcs ou plus font défaillance, l'on perd toute la demande du réseau.

Une façon d'améliorer la borne supérieure (5.12) serait alors d'évaluer une borne supérieure du refus pour les états de panne double. L'équation (5.13) donne une idée de la forme de la borne:

$$E[Z]_{sup} = P(S_1)Z(S_1) + \sum_{k=2}^{m} P(S_k)\cdot Z(S_k) \\ + \sum_{j\in D_2} P(S_j)\hat{Z}(S_j) + \\ + \sum_{j\in D_3} P(S_j)Z(S_{2^n}). \qquad (5.13)$$

Dans l'équation précédente, D_2 dénote l'ensemble des indices des états avec deux arcs en panne et D_3 celui des indices des états avec trois arcs ou plus en panne. La notation $\hat{Z}(S_j)$ indique une borne supérieure à la valeur $Z(S_j)$.

Nous avons donc répondu partiellement à la question a) de la section précédente: il y a au moins une meilleure borne supérieure au refus. Mais comment garantir qu'avec le facteur combinatoire à ajouter, elle soit trouvable sans trop ralentir l'algorithme? La réponse est basée, évidemment, dans la façon de calculer les $\hat{Z}(S_j) \forall\, j \in D_2$.

Une $\hat{Z}(S_j)$ qui semble correcte est la valeur du refus total calculé lorsqu'un seul des arcs défaillant est actuellement en panne plus la valeur de la demande du deuxième arc défaillant. Par exemple: si S_{70} correspond à l'état "l'arc 10 et 12 sont en panne", alors $\hat{Z}(S_{70}) = Z(S_{10}) + X_{12}(S_{10})$ (Note: par abus de notation, on suppose que l'état S_{10} équivaut à l'état "l'arc 10 fait défaillance") $X_{12}(S_{10})$ dénote la demande qui essaie de passer sur l'arc 12 lorsque l'on est dans l'état S_{10}.

Le calcul des $\hat{Z}(S_j)$ défini dans le paragraphe précédent nécessiterait uniquement des résultats du routage des pannes simples, qu'on a déjà, et de plusieurs additions. Le temps de calcul ne serait donc pas trop affecté. Cependant, un obstacle s'impose: nous n'avons pas la garantie que le $\hat{Z}(S_j)$ tel que défini soit effectivement une borne supérieure au $Z(S_j)$. En effet, lorsque le réseau en état de fonctionnement parfait est déjà très congestionné, le reroutage du flot sur l'arc en panne pourrait causer une plus grande congestion sur les arcs voisins de façon à avoir plus de refus avec que sans reroutage. Ce paradoxe est expliqué par l'erreur introduite par la forme de la fonction objectif utilisée dans l'optimisation du routage. En effet, le fait d'avoir une fonction qui est séparable par arcs implique, indirectement, que le refus pour une route pour une paire O/D équivaut à la somme des refus sur tous les arcs de la route. Ceci indique, de façon indirecte aussi, que l'on néglige les termes croisés de probabilités de refus.

Si le niveau des demandes et des capacités du réseau est tel que les termes croisés des probabilités de refus sont assez petits, le reroutage du flot sur un arc en panne sera toujours préférable à la perte de la demande de l'arc.

Puisque notre réseau de départ était assez congestionné, même en état de fonctionnement parfait, nous avons procédé à la construction d'un nouveau réseau. Le conception du nouveau réseau était réalisée en trois pas:

1) Réaliser le routage optimal pour l'état de fonctionnement parfait.

2) Identifier les arcs pour lesquels la fonction objectif a une pente de un.

3) Augmenter la capacité des arcs identifiés dans (2) jusqu'a la valeur du flot optimal sur ces arcs.

Le deuxième point mérite une justification. Pour réduire les termes croisés de probabilités de refus il faut réduire les probabilités de refus elles-mêmes. Puisque la probabilité de refus sur un arc est exprimée par la valeur de l'objectif (l'espérance du trafic perdu sur l'arc) divisée par le trafic offert à l'arc, il est évident qu'une diminution de l'espérance du trafic perdu entraîne une diminution de la probabilité de refus. A demande fixe, l'espérance du trafic perdu est diminuée uniquement par redimensionnement des arcs et, évidemment, les premiers arcs à être redimentionnés seront ceux qui présentent la plus grande probabilité de refus. Une façon facile d'identifier ces arcs est de vérifier si la pente de la fonction objectif est l'unité.

Nous avons effectué les opérations (1) à (3) sur notre réseau de départ jusqu'à l'obtention d'un réseau qui, pour la matrice de demande donnée, ne présentait pas beaucoup de congestion. Ensuite, nous avons trouvé la borne supérieure du refus sans réoptimisation et les bornes supérieures et inférieures avec réoptimisation. Notons que la borne supérieure sans réoptimisation a été calculée en considérant que toute la demande d'un arc en panne est perdue. Les résultats sont montrés dans les figures 5.13 et 5.14.

La figure 5.13 montre que si le réseau a été bien conçu, il est toujours préférable de rerouter la demande qui serait normalement offerte aux arcs en panne plutôt que de la perdre. La figure 5.14 indique que la distance qui existe entre les bornes supérieures du refus sans et avec réoptimisation croît de façon linéaire avec le temps de défaillance moyen du réseau.

Nous avons, jusqu'ici, répondu aux questions (a) et (c) posées dans la section précédente. À l'aide de la figure 5.13, il est aussi possible de répondre à la question: comment diminuer le temps de calcul tout en ayant une borne de refus comparable. En effet, si l'on a un réseau à faible congestion et que le temps moyen de défaillance par année est petit, l'on pourrait utiliser comme borne supérieure à l'espérance du refus la borne supérieure du refus sans réoptimisation. Il serait alors possible de trouver un résultat proche de celui voulu dans un temps d'exécution qui est de l'ordre de n (le nombre d'arcs) fois plus court que le temps nécessaire à la réoptimisation du refus, lorsque l'on considère seulement les états de pannes simples.

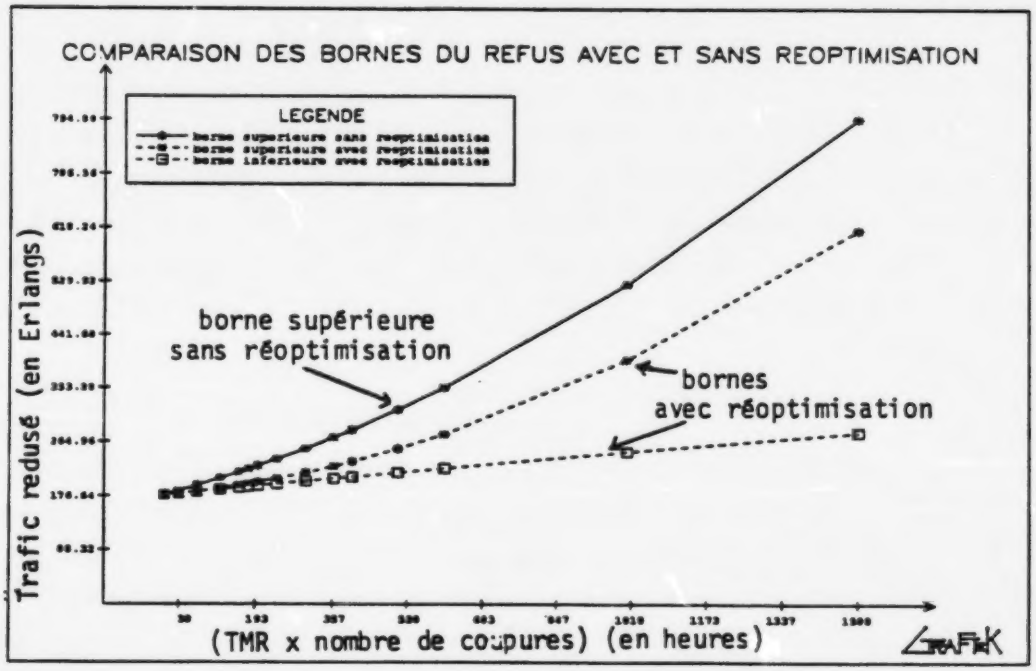


Figure 5.13 Comparaison des bornes du refus avec et sans réoptimisation.

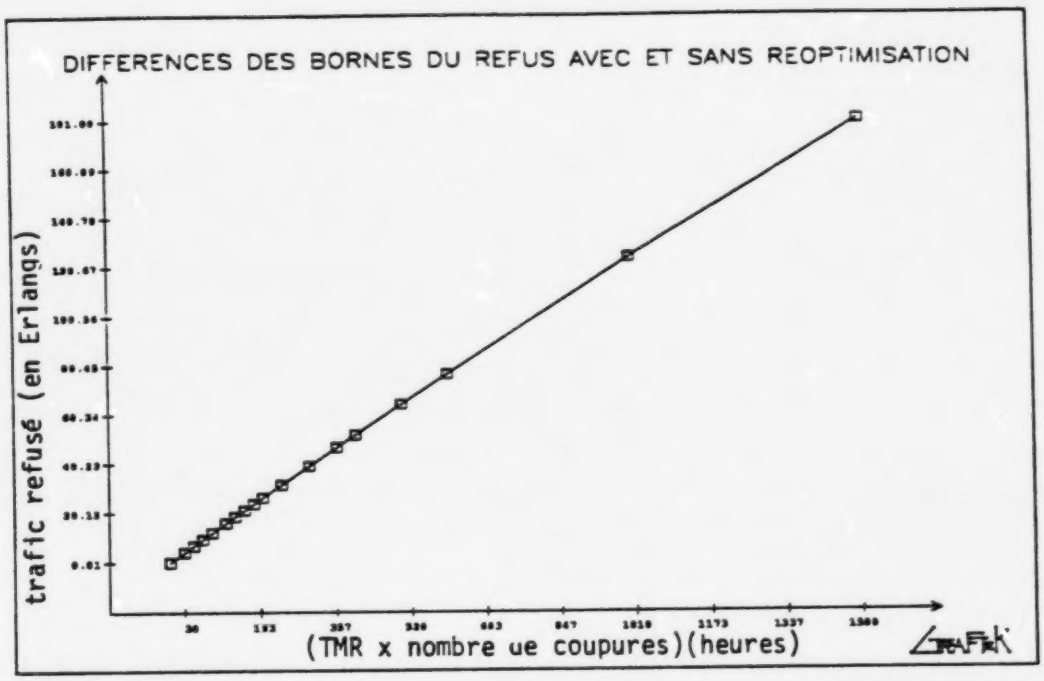


Figure 5.14 Différences entre les bornes supérieures du refus avec et sans réoptimisation.

CONCLUSION

Le but principal de ce travail était de définir une bonne mesure de fiabilité et de trouver une méthodologie pour calculer cette mesure pour des réseaux de taille réelle. Nous avons produit un prototype modèle-algorithme qui calcule une mesure de fiabilité en se basant sur la notion de routage optimal. Ce prototype évalue la performance du réseau et, en plus, identifie ses points faibles. Or il serait possible de l'utiliser non seulement pour mesurer la fiabilité d'un réseau de télécommunications mais, aussi, pour évaluer des nouvelles politiques de routage qui pourraient avoir une meilleure performance du réseau comme objectif.

Pour ce qui est de l'originalité de la méthode ainsi que des apports réalisés dans ce travail, nous avons conçu pour la première fois une méthodologie pour évaluer une mesure de fiabilité de réseaux reliée à la notion de routage et calculable en temps non exponentiel. La deuxième contribution de cette étude était la démonstration de la convexité de la fonction espérance des pertes d'Erlang par rapport à la demande. Ceci constitue un résultat majeur en télécommunication car, la conjecture de la convexité de la fonction était acceptée mais, à notre connaissance, n'avait pas été prouvée. Troisièmement, nous avons montré une borne supérieure de l'espérance du refus dans le réseau qui est plus serré que la borne de Li et Silvester.

En ce qui concerne la méthodologie proposée, les figures de la Section 5.4 nous permettent d'affirmer que, lorsque les taux de panne correspondent à des valeurs comparables à celles que l'on a considérées, l'effort sur la précision de l'algorithme de routage donne de meilleurs résultats qu'un effort concentré sur la plage des états de panne prise en considération. Nous pouvons alors conclure que le modèle ainsi que la politique de routage utilisés constituent les élements les plus importants dans l'évaluation de la fiabilité d'un réseau existant. Il faut, cependant, souligner que cette conclusion est valide uniquement si le réseau en étude est congestionné même en état de fonctionnement parfait. Si, par contre, l'on étudie la fiabilité d'un réseau surdimensionné ou n'ayant pas de pertes dans l'état de fonctionnement, les états de pannes prennent plus d'importance et l'on doit concentrer les efforts sur l'algorithme d'énumération partielle d'états ainsi que sur l'étude de meilleures bornes.

Dans cette optique, nous jugeons que le travail futur à réaliser en fiabilité de réseaux existants doit s'appuyer sur l'amélioration des modèles de routage plutôt que sur l'implantation d'algorithmes d'énumération pouvant reserrer les bornes de l'espérance du refus. Les modèles de routage existants actuellement dans les réseaux de télécommunications obéissent à des règles de commutation plutôt qu'à une politique d'optimisation véritable. Nous avons proposé des modèles d'optimisation d'acheminement d'appels sous deux types de politiques: centralisée et décentralisée. Le routage actuel dans des réseaux à commutation par circuits s'apparente davantage à nos modèles d'optimisation décentralisée. Cependant, le développement rapide des technologies en télécommunications nous permet de prévoir que dans un futur proche, il sera possible de réaliser une commutation d'appel s'approchant de l'optimisation globale. Nous avons donc implanté ce type de modèle qui reflète un routage beaucoup plus efficace que celui qui obéit à une politique décentralisée. Il serait néanmoins intéressant d'implanter les modèles d'optimisation égoïste proposés dans le chapitre quatre afin d'évaluer la performance du réseau sous une politique d'optimisation beaucoup plus proche des règles de routage actuelles. Pour ce qui est du modèle d'optimisation globale employé, nous mentionnons qu'il serait possible de l'améliorer en le rendant plus réaliste. Le modèle, tel qu'il a été présenté dans ce travail, utilise des valeurs moyennes de demande alors que dans la réalité, même dans un intervalle de temps restreint, il peut exister une variation assez importante de la demande d'appels. Il serait donc intéressant de modifier la fonction objectif pour qu'elle prenne en considération la variation de la demande. Cette modification rendrait le modèle plus réaliste et réduirait la courbure de la fonction de perte au voisinage de la capacité de l'arc ce qui améliorerait la performance de l'algorithme de résolution.

Pour ce qui est de l'algorithme de résolution du problème de routage, il est évident que l'algorithme de Frank-Wolfe n'est pas celui qui produit les meilleures performances, ceci dû au fait qu'il ne réoptimise pas de façon efficace. Nous suggérons les méthodes suivantes: 1) l'implantation d'un algorithme de Frank-Wolfe accéléré, tel que la méthode de PARTAN, pour que l'algorithme de routage, dans tous les états considérés, converge plus rapidement; 2) l'implantation d'une modification de la variante "away-step" de Frank-Wolfe pour les états de panne de

façon à ce que dans les itérations réalisées avec cette variante le flot sur les arcs en panne puisse être soustrait; 3) l'utilisation du simplexe convexe qui permetterait de réoptimiser efficacement les états de panne, l'on ferait demarrer l'algorithme à partir de la solution optimale de l'état de fonctionnement parfait tout en pénalisant fortement le passage de flot sur l'arc en panne.

Bien que le prototype que nous avons créé ait été employé principalement dans l'évaluation de la performance d'un réseau existant, nous avons donné une esquisse d'utilisation en conception de réseau. La capacité des arcs n'étant plus une contrainte infranchissable dans la conception de réseaux, à cause de l'introduction de liaisons à fibre optique, les contraintes de fiabilité gagnent énormément d'importance. Il serait possible d'utiliser la méthodologie décrite dans ce travail pour faire une conception qui considère, en plus du facteur capacité des arcs, d'autres facteurs reliés à la performance du réseau. Des paramètres tels que le nombre d'employés ou d'équipements nécessaires à la réparation des pannes pourraient être introduits.

En effet, les figures 5.9 à 5.13 indiquent que l'espérance du refus augmente avec le produit du nombre de coupures par temps moyen de réparation. Notre prototype permet d'étudier des scénarios où il serait possible d'obtenir le même niveau de refus en diminuant le nombre de coupures ou le temps moyen de réparation. Il serait donc possible pour la compagnie de télécommunications de mieux quantifier et de prévoir la productivité de son service de réparations. Un des travaux à réaliser à partir des modèles ici présentés est la création d'heuristiques, plus fines que celle que l'on a exquissée, qui font varier plusieurs paramètres tout en gardant comme base le prototype d'évaluation de la performance du réseau.

Pour terminer, nous mentionnons que même si la méthodologie proposée a été conçue pour un réseau de télécommunications, il est possible de la modifier afin d'évaluer la performance ou de concevoir d'autres types de réseaux de flot.

BIBLIOGRAPHIE

K.K. AGGARWAL, K.B. MISRA, A Fast Algorithm for Reliability Evaluation, *IEEE Transactions on Reliability*, Vol. R-24, avril 1975, pp. 83–85.

K.K. AGGARWAL, J.S. GUPTA, K.B. MISRA, A Simple Method for Reliability Evaluation of a Communication System, *IEEE Transactions on Communications*, mai 1975, pp. 563–566.

K.K. AGGARWAL et S. RAI, Reliability Evaluation in Computer-Communication Networks, *IEEE Transactions on Reliability*, Vol. R-30, No 1, avril 1981, pp. 32–35.

K.K. AGGARWAL, Y.C. CHOPRA, J.S. BAJWA, Capacity Consideration in Reliability Analysis of Communication Systems, *Transactions on Reliability*, Vol. R-31, juin 1982, pp. 177–181.

K.K. AGGARWAL, Integration of Reliability and Capacity in Performance Measure of a Telecommunication Network, *IEEE Transactions on Reliability*, Vol. R-34, No 2, 1985, pp. 184–186.

A.A. ASSAD, Multicommodity Network Flows - A Survey, *Networks*, Vol. 8, 1978, pp. 37–91.

M.O. BALL, Computing Network Reliability, *Operations Research*, Vol. 27, No 4, juillet-août 1979, pp. 823–838.

P. BARAN, On Distributed Communication Networks, *IEEE Transaction on Communication System*, Vol. CS-12, mars 1964, pp.1–9.

D. BEAR, Principles of Telecommunication Traffic Engineering, *IEE Telecommunication Series*, 1980.

F.T. BOESCH et R.E. THOMAS, On Graphs of Invulnerable Directed Communication Networks, *IEEE Transactions on Communication Technology*, Vol. COM-18, octobre 1970, pp. 484–489.

F.T. BOESCH, Network Reliability, the State of the Art Foreword, *IEEE Transactions on Reliability*, Vol. R-35, No 3, août 1986, pp. 229–239.

V. BONAVENTURE, S. CACOPERDI, M. DÉCINA et A. NOVEN, Service Availability of Communication Networks, IEEE 1980, National Telecommunications Conference, pp. 15.2/1–6.

J.K. CAVERS, Cutset Manipulations for Communication Network Reliability Estimation, *IEEE Transactions on Communications*, Vol. COM-23, No 6, juin 1975.

B.N. CLARK, E.M. NEWFELD, C.J. COLBOURN, Maximizing the Mean Number of Communicating Vertex Pairs in Series-parallel Networks, *IEEE Transactions on Reliability*, Vol. R-35, août 1986, pp. 247–251.

P. DOULLIEZ et E. JAMOULLE, Transportation Networks with Random Arc Capacities, *Revue Française d'Automatique, Informatique et Recherche Opérationnelle*, novembre 1972.

W. FELLER, An Introduction to Probability Theory and its Applications, Vol. I, 3ième édition, p. 99, 1968.

G.S. FISHMAN, A Monte Carlo Sampling Plan for Estimating Parameters and Related Functions, University of North Carolina at Chapel Hill, Technical Report, No UNC/ORSA/ TR–85/7, juin 1985.

M. FLORIAN, An Introduction to Network Models Used in Transportation Planning, publication # 262, Centre de Recherche sur les Transports, août 1982.

L.R. FORD, et D.R. FULKERSON, Flots dans les graphes, Gauthier-Villars éditeur, Paris, 1974.

H. FRANK, Vulnerability of Communication Networks, *IEEE Transaction on Communication Technology*, Vol. COM–15, décembre 1967, pp. 778–789.

H. FRANK, Survivability Analysis of Command and Control Networks - Part I, *IEEE Transactions on Communications*, Vol. COM-22, No 5, mai 1974A, pp. 589–595.

H. FRANK, Survivability Analysis of Command and Control Networks - Part II, *IEEE Transactions on Communications*, Vol. COM-22, No 5, mai 1974B, pp. 596–605.

J. FRATTA et U.G. MONTANARI, A Boolean Algebra Method for Computing the Terminal Reliability in a Communication Network, *IEEE Transactions on Circuit Theory*, Vol. CT-20, No 3, mai 1973, pp. 203–211.

A. GIRARD, Routing and Dimensionning in Circuit Switched Networks, Addison Wesley éditeurs, à paraître.

M. GONDRAN et M. MINOUX, Graphes et algorithmes, *Collection de la Direction des Études et Recherches d'Électricité de France*, Eyrolle éditeur, 1979.

J. GUÉLAT, Algorithmes pour le problème d'affectation du trafic d'équilibre avec demandes fixes comparaisons, publication #299, Centre de Recherche sur les Transports, mai 1983.

J.N. HAGSTROM, Using the Decomposition-Tree of a Network in Reliability Computation, *IEEE Transactions on Reliability*, Vol. R-32, avril 1983, pp. 71–78.

E. HANSLER, G.K. McAULIFFE, R.S. WILKOV, Optimizing the Reliability in Centralized Computer Networks, *IEEE Transactions on Communications*, Vol. COM-20, juin 1972, pp. 640–644.

E. HANSLER, G.K. McAULIFFE, R.S. WILKOV, Exact Calculation of Computer Network Reliability, *Networks*, 4, 1974, pp. 95–112.

J.E. HOPCROFT et R.E. TARJAN, Dividing a Graph into Triconnected Components, *SIAM J. Comput.* Vol. 2, no 3, septembre 1973, pp. 135–158.

K. KORTANEK, G. POLAK, Network Design and Dynamic Routing under Queueing Demand, Carnegie-Mellon University, Working Paper, 1984.

Y.F. LAM et V.O.K. LI, On Network Reliability Calculations with Dependent Failures, *IEEE Global Telecommunications Conference*, Conference Record, Vol. 3, 1983, pp. 1499–1503.

S. LAVOIE, Modèles de Comportement des Passagers en Transport Aérien, publication # 408, Centre de Recherche sur les Transports, juin 1984.

S.H. LEE, Reliability Evaluation of a Flow Network, *IEEE Transactions on Reliability*, Vol. R-29, No 1, avril 1980, pp. 24–26.

V.O.K. LI et J.A. SILVESTER, Performance Analysis of Networks with Unreliable Components, *IEEE Transactions on Communications*, Vol. COM-32, No 10, octobre 1984, pp. 1105–1110.

E.M. NEUFELD et C.J. COULBOURN, The Construction of Reliable Series-Parallel Networks, *Congressus Numerantium*, Vol. 41, 1984, pp. 21–26.

S.N. PAN et J.D. SPRAGINS, Dependent Failure Reliability Models for Tactical Communications Networks, *IEEE International Conference on Communications: Integrating Communication for World Progress*, Vol. 2, 1983, pp. 765–771.

T. POLITOF et A. SATYANARAYANA, Efficient Algorithms for Reliability Analysis of Planar Network - A Survey, *IEEE Transactions on Reliability*, Vol. R-35, No 3, août 1986, pp. 252–259.

S. PROVAN et M.O. BALL, Computing Network Reliability in Time Polynomial in the Number of Cuts, *Operations Research*, Vol. 2, No 3, mai–juin 1984, pp. 516–526.

S. RAI et K.K. AGGARWAL, An Efficient Method for Reliability Evaluation of a General Network, *IEEE Transactions on Reliability*, Vol. R-27, No 3, août 1978, pp. 206–211.

A. SATYANARAYANA et R.K. WOOD, A Linear-Time Algorithm for Computing K-Terminal Reliability in Series-Parallel Networks, *SIAM J. Comput.*, Vol. 14, No 4, novembre 1985, pp. 818–832.

M. SAUVE, Résolution numérique du problème de multiflot compatible à coût minimum, publication #409, Centre de Recherche sur les Transports, octobre 1984.

A.W. SHOGAN, Modular Decomposition and Reliability Computation in Stochastic Transportation Networks Having Cutnodes, *Networks*, Vol. 12, 1982, pp. 255–275.

F. SOUMIS, Planification d'une flotte d'avions, Publication #318, Centre de Recherche sur les Transports, décembre 1978.

J. SPRAGINS, Reliability Problems in Data Communications Systems, Proceedings, Fifth ACM/IEEE Data Communications Symposium, septembre 1977.

J. SPRAGINS, Dependent Failures in Data Communication Systems, *IEEE Transactions on Communications*, Vol. COM–25, décembre 1977, pp. 1494–1499.

J. SPRAGINS, Limitations of Current Telecommunication Network Reliability Models, *IEEE Global Telecommunications Conference*, Conference Record, Vol 2, 1984, pp. 836–840.

J. SPRAGINS, A Fast Algorithm for Computing Availability in Networks with Dependent Failures, Proceedings of the IEEE Infocom 84, 1984, pp. 36–42.

J. SPRAGINS et J. ASSIRI, Communication Network Reliability Calculation with Dependent Failures, Conference Record 1980 National Telecommunications Conference, Houston, Texas, décembre 1980.

R. VAN SLYKE et H. FRANK, Network Reliability Analysis: Part I, *Networks*, 1, pp. 279–290, 1972.

J.A. WALD et C.J. COULBOURN, Steiner Trees in Probabilistic Networks, *Microelectronics and Reliability*, Vol. 23, No 5, 1983, pp. 837–840.

R.S. WILKOV, Anaiysis and Design of Reliable Computer Networks, *IEEE Transaction on Communications*, juin 1972, pp. 660–677.

R.K. WOOD, Factoring Algorithms for Computing K-Terminal Network Reliability, *IEEE Transactions on Reliability*, Vol. R-35, No 3, août 1986, pp. 269–278.